

PROCÉDURES OPTIMALES POUR LE CLASSEMENT DES t MEILLEURS ARTICLES PARMIS n AU MOYEN DE COMPARAISONS BINAIRES

par Milton SOBEL ⁽¹⁾

Traduction & adaptation ⁽²⁾

Alain LAHOUE, Gérard LOUIT, Jean-Pierre ROUBELAT ⁽³⁾

Résumé. — On veut ordonner les t plus grands nombres parmi n nombres tous différents par le seul moyen de comparaisons binaires. Le but essentiel est de trouver une procédure minimisant la longueur de cheminement du questionnaire de tri; un autre but est de trouver une procédure minimisant le nombre maximal de tests exigés pour effectuer ce classement. L'apport essentiel de cette étude repose sur la notion de couplage et sur l'évaluation de l'espérance mathématique de l'incertitude. Une procédure semblant s'imposer est déduite de la considération de la maximisation de l'information apportée localement par toute comparaison.

AVERTISSEMENT

La traduction de certains termes méritait une périphrase d'explication; aussi en avons-nous proposé une traduction correspondante approximative. Nous indiquons ici les termes originaux et leurs traductions approchées.

Knock-out tournament	tournoi «K.O.».
Binary errorless comparison	comparaison entre deux éléments à une alternative.
«E-goal», «M-goal»	«E», «M»; (cf. chapitre 1).
Maximum length	longueur du plus grand chemin.
One step entropy procedure	procédure entropique à un rang.
Fully-inductive	récurrente: $P(n) = fP(n-1)$.
Semi-inductive	semi-récurrente: $P(n) = fP(\frac{n}{2})$
Ordinary pairing	couplage ordinaire.
Cycle pairing	couplage cyclique.
Complete pairing	couplage total.
Second best	sous-vainqueur.
«a vs. b»	comparaison, match, test, entre a & b.

¹Université du Minnesota.

²Original: On an optimal search using binary errorless comparisons, oct. 1968, Minneapolis.

³Équipe de Recherche «Structures de l'information», C.N.R.S., Paris.

CHAPITRE 1

INTRODUCTION

On veut ordonner les t plus grands nombres parmi n nombres tous différents, par le seul moyen de comparaisons binaires (deux à deux); il peut s'agir également de ranger des objets selon un critère de poids associé à chacun d'entre eux.

Dans certaines applications, on ne connaît pas ces nombres, dans d'autres, par exemple le problème du tri, ils sont connus effectivement.

EXEMPLES:

— Une machine ou une personne peut avoir à ordonner de façon croissante n nombres initialement distribués au hasard en n'utilisant que des comparaisons binaires.

— Le problème du tournoi: il faut classer n joueurs de tennis selon leur force. On suppose que les joueurs sont de valeurs différentes, que le meilleur gagne toujours, et que la relation «plus fort que» est transitive.

— Le problème de pesée: on dispose de n poids inégaux et d'une balance qui n'autorise qu'une seule masse sur chaque plateau; il faut ordonner ces poids.

En termes de théorie des questionnaires (qui réunit à la fois les points de vue de la théorie des graphes et de la théorie de l'information pour le problème considéré) on parle de problèmes de tri.

Pour nous, tous ces aspects peuvent se formuler de façon unique en faisant

$$t = n \quad (\text{ou } t = n - 1);$$

nous préférons appeler cela le «problème d'espérance de Steinhaus», pour $t = n - 1$ en raison de l'intérêt que Steinhaus porta très tôt au problème du minimax (voir plus loin).

Au départ, on suppose que les n nombres sont en ordre aléatoire, soit qu'on ait rendu cet ordre aléatoire soit qu'on puisse le supposer tel.

Pour expliquer notre objectif considérons le nombre T de comparaisons binaires (ou tests) requis pour $n = 3$. T peut valoir 2 ou 3 et à partir de la répartition aléatoire initiale, on obtient, pour la procédure optimale, l'espérance du nombre de tests ou longueur de cheminement: $E\{T|n=3\} = \frac{8}{3}$.

— Notre but essentiel est de trouver une procédure R qui minimise cette longueur de cheminement. Dans cette étude on introduit plusieurs nouvelles procédures ayant toutes des longueurs de cheminement plus petites que celle obtenue par la procédure de Steinhaus définie plus loin. Certaines d'entre elles ont des longueurs de cheminement plus petites que celles de toutes les procédures que nous connaissons et parmi celles-là nous pouvons supposer qu'il en existe d'optimales.

— Un autre but de cette étude est de trouver une procédure R qui minimise le nombre maximal de tests exigés pour ordonner les t plus grands de n nombres; à cet effet nous nous référons au «problème du minimax de Steinhaus», ou de longueur minimale du plus long chemin.

Ces deux buts ne sont pas sans rapport et pour de petites valeurs de n nous pourrions trouver des procédures réalisant à la fois la minimisation de la

longueur de cheminement (notées E -optimales) et l'obtention du plus petit maximum (notées M -optimales).

Dans l'optique du minimax, Steinhaus [23] donne une procédure fondamentale R_S , récurrente. En 1950 lors de l'édition de son ouvrage, il conjecturait que cette procédure était optimale pour toute valeur de n ; mais dans un autre ouvrage [24] d'édition plus récente il abandonne ce point de vue et donne un contre-exemple dans le cas où n vaut 5.

Bien que la procédure R_S soit en queue de notre gamme de procédures pour $t = n - 1$ (plus grande longueur de cheminement et maximum le plus élevé dans le tableau du chapitre 5), elle constitue une base de comparaison de deux façons:

— D'une part parce que parmi les procédures récurrentes elle est à la fois E -optimale et M -optimale [10].

— D'autre part parce qu'elle est connue pour beaucoup de ses propriétés.

Pour cette procédure Kisilitsyn trouva des bornes générales de la longueur de cheminement [14]; par passage à la limite, il en déduisit une expression asymptotique de la longueur de cheminement [15]. Cette procédure est bien connue en informatique; on l'appelle «insertion binaire» ou «TID», «rangement par insertion» ou «recherche binaire», selon les auteurs. Malgré cela, il est étonnant de voir combien d'auteurs supposent implicitement ou explicitement (cf. page 236 de [13]), que R_S est soit E -optimale, soit M -optimale, ou les deux sans avoir étudié d'autres travaux.

Des deux points de vue « E » et « M », une autre procédure importante est la procédure «semi-récurrente» R_F de Ford et Johnson [8], bien que l'étude en question, ne concerne que le problème du minimax.

En fait, la procédure R_F est E -optimale pour $n \leq 5$ et les valeurs de la longueur de cheminement pour de faibles valeurs de n , (calculées par A. Hadian et l'auteur) se sont avérées plus petites que toutes celles que l'on a pu faire paraître en littérature jusqu'au début de cette étude. Cesari [4] et Hadian [10] ont modifié la procédure R_F pour $n \geq 6$ afin d'obtenir une longueur de cheminement plus petite sans changer la valeur pour « M ».

Picard [17] a donné une procédure pour $n = 6$ (et $t = 5$) qui est à la fois E -optimale et M -optimale. L'approche qu'il fait du problème au moyen de la théorie des questionnaires lui permet de l'appliquer à de nombreux et intéressants problèmes de recherche.

Pour être complet, nous devons également mentionner les travaux de Bose et Nelson [1], Hibbard [11].

Nous n'en ferons pas état dans nos analyses car:

- elles nécessitent d'introduire des restrictions sur l'encombrement mémoire,
- leur critère de comparaison diffère sensiblement du nôtre (nombre T de comparaisons binaires), ou
- leurs résultats ne sont pas directement comparables aux nôtres.

Aussi, notre problème est-il lié à celui de la fusion de séquences ordonnées de nombres en une seule, à condition que le critère utilisé soit uniquement le nombre T de comparaisons binaires requises et non comme le fait Burge [2], le nombre total de transferts de clés.

On peut observer empiriquement que du point de vue de Burge, nos procédures donnent d'aussi bons résultats que la sienne mais que, par contre, elles lui sont supérieures du point de vue de notre critère.

L'apport essentiel de notre étude est fondé sur deux idées: le «couplage» et l'évaluation de l'espérance mathématique de l'incertitude. Notre procédure R_E sélectionne à chaque étape le test qui maximise l'espérance de la réduction d'entropie due à une comparaison unique.

De façon équivalente, elle détermine le test qui donne la plus faible valeur d'incertitude (ou apporte le maximum d'information). En introduisant certains types de couplage pour les premiers tests, on peut simplifier notablement la procédure et, dans certains cas, l'améliorer réellement. L'idée de l'espérance de l'entropie fut utilisée pour le problème de test de groupe (Groll et Sobel [22]) et par F. Dubail [7] dans d'autres problèmes de recherche, d'«Entropie généralisée».

Notre principal intérêt réside dans les procédures entropiques à un rang. Comme dans [22], pour le problème de test de groupe, on peut considérer une généralisation évidente de R_E , appelée $R_{E,g}$ qui détermine le test maximisant l'espérance de la réduction d'entropie des $g(\geq 1)$ tests suivants. Toutes les procédures sont telles qu'elles peuvent tenir compte de la répartition initiale aussi bien que de l'information acquise à chaque étape.

La procédure $R_E = R_{E,1}$ (type de procédure à un rang) fournit des valeurs optimales de la longueur de cheminement pour de petites valeurs de n ($n \leq 6$ pour $t = 2$ et $t = n - 1$) partout où on peut trouver des procédures optimales.

De plus, chacune des trois procédures entropiques améliore considérablement les résultats connus pour de faibles valeurs de n . En fait, il s'avère intéressant de trouver des exemples où R_E ne serait pas optimale. Tous nos résultats empiriques confirment l'hypothèse qu'une procédure E -optimale peut être obtenue à partir d'une procédure R_{E_1} ou de la famille $R_{E,g}$ pour de faibles valeurs de g .

Nous étudierons en premier lieu le cas $t = 2$ et ensuite le cas $t = n - 1$; d'une part parce que le problème est plus simple et d'autre part cela permet de montrer en même temps la complexité du cas général ($1 \leq t \leq n - 1$).

Le cas des petites valeurs de t s'apparente étroitement, en son début, à l'étude de Lewis Carroll [3] qui traite de la manière «erronée» (Système de coupe) d'accorder le deuxième prix au cours d'un tournoi de tennis.

Il fit remarquer que si les joueurs sont éliminés après une défaite, il y a alors de fortes chances de ne pas trouver l'authentique «sous-vainqueur».

Par exemple, pour $n = 8$ en utilisant le couplage complet (ou tournoi «K.O.» opposant par paire tous ceux qui n'ont pas perdu), le «sous-vainqueur» à la probabilité $\frac{3}{7}$ d'être dans le groupe des 4 compétiteurs contenant le vainqueur et, partant, de ne pas recevoir le 2^e prix.

Steinhaus [23] a discuté le cas où $t = 2$; les rapports de J. Schreier [19] et J. Ślupcecki [20] montrent la validité de notre conclusion sur deux de nos procédures: elles sont M -optimales pour $t = 2$. C. F. Picard [17] a étudié aussi le cas $t = 2$, et une de ses procédures, R_P figure à notre tableau de comparaison. Nous considérons la procédure R_P , pour $t = 2$, analogue à la procédure R_S de Steinhaus pour $t = n - 1$; nous ne retenons d'ailleurs que les procédures au moins aussi bonnes que R_P des deux points de vue « E » et « M ».

Les travaux de David [5], Glenn [9] et Maurice [16] qui traitent des tournois «K.O.» s'apparentent à notre sujet mais pas avec cette étude-ci. Ils considèrent, eux, aléatoire de supposer qu'un joueur plus fort qu'un autre ait une plus grande

probabilité de le vaincre. Dans notre cas, le meilleur joueur gagne toujours et le hasard n'intervient seulement que dans la répartition initiale des n joueurs. On conçoit que, connaissant les meilleures procédures qui s'appliquent dans ce dernier cas, on pourrait en déduire, avec profit, des procédures s'appliquant à des cas où le hasard intervient dans les résultats. David [6] a donné une critique précise des travaux de David, Glenn et Maurice.

Bien qu'on n'ait pas essayé ici d'envisager de grandes valeurs de n , ni de trouver les meilleures procédures pour le calcul sur machine, nous pensons que notre procédure entropique, ou toutes celles qui en dérivent, peuvent être compétitives de ces points de vue. On suppose que les résultats seront bien meilleurs que tous ceux déjà publiés (cf. Bose et Nelson [1]) même si pour comparer les procédures on utilise un critère autre que le nombre de comparaisons.

Avant de poursuivre plus avant, nous donnons au préalable quelques définitions, qui seront plus amplement explicitées au chapitre 4.

— *Couplage ordinaire*

Lorsque $n = 2k$ ou $2k + 1$, on fait un couplage ordinaire lorsqu'on associe par deux les $2k$ premiers joueurs (ou items) et que l'on effectue alors seulement k comparaisons (laissant éventuellement tomber le $2k + 1^{\text{ième}}$).

— *Couplage cyclique*

Lorsque $n = 2^r + c$, on associe par paire les joueurs de l'ensemble " 2^r " et on trouve le vainqueur en effectuant $2^r - 1$ comparaisons.

— *Couplage total*

Lorsque $n = \sum_{i=1}^s 2^{r_i}$, on fait les p (cf. chapitre 4) comparaisons pour trouver le vainqueur de chaque sous-ensemble r_i .

— *Tournoi K.O. (Knock Out)*

Pour $n = 2^r$, on fait un couplage ordinaire sur tous les vainqueurs du tour précédent. On détermine donc le vainqueur en r tours et en $2^r - 1$ comparaisons.

CHAPITRE 2

PROCÉDURES POUR LE PROBLÈME DE RANGEMENT AVEC $t = 2$

Nous présentons ici plusieurs procédures: toutes sont nouvelles, sauf celle de C. F. Picard (notée R_P).

L'une d'entre elles, notée R_F , est une adaptation de la procédure de Ford et Johnson pour $t = 2$.

R_{E_1} , l'une des procédures entropiques, est de façon générale aussi bonne ou meilleure que toute autre procédure pour toutes les valeurs considérées, $2 \leq n \leq 10$.

D'après les travaux de J. Schreier [19] et Ślupecki [20], deux de ces procédures sont M -optimales. Chacune d'entre elles est décrite brièvement dans ce chapitre et nous donnons un tableau comparatif des résultats pour chaque valeur de n . Propriétés et dérivations des résultats sont données dans le chapitre suivant.

Nous utilisons le terme «récurrent» pour signifier que l'algorithme pour n joueurs se déduit de celui pour $n - 1$; l'expression «semi-récurrent» signifie que l'algorithme pour n joueurs se déduit directement de celui pour $\lfloor \frac{n}{2} \rfloor$ joueurs, où « $\lfloor x \rfloor$ » désigne le plus grand entier $\leq x$.

Tous les logarithmes sont à base 2, sauf indication contraire.

Procédure R_E

C'est la procédure entropique à un rang pour $t = 2$; le principe de base est de trouver le test qui maximise l'espérance de la réduction d'entropie après une comparaison.

Procédure R_{E_1}

On suppose que $n = 2^r + c$ ($0 \leq c < 2^r$); la procédure débute par un tournoi «K.O.» effectué sur 2^r joueurs et on achève le problème par la méthode d'entropie à un rang.

Pour un couplage total et pour $n = 2^r + c$, nous nous autorisons le couplage parmi les c joueurs restants; nous écrirons alors n la forme:

$$n = 2^{r_1} + 2^{r_2} + 2^{r_3} + \dots + 2^{r_s}$$

et $r_1 > r_2 > r_3 > \dots > r_s \geq 0$, et réalisons un tournoi «K.O.» pour chaque puissance de 2.

Procédure R_{E_2}

Pour cette procédure nous effectuons un couplage total et achevons au moyen de la méthode d'entropie à un rang.

Procédure R_F

Cette procédure est analogue à celle de Ford et Johnson appliquée au cas $t = 2$.

Supposons que n vaille $2k$ ou $2k + 1$, la procédure se fait en trois étapes:

- 1° Utilisant un couplage ordinaire, nous prenons $2k$ joueurs sur qui nous effectuons k comparaisons, laissant de côté un joueur dans le cas où n est impair.
- 2° Par récurrence, nous sélectionnons les 2 meilleurs joueurs parmi les k vainqueurs de l'étape n° 1 (la procédure est évidente pour $n = 2$ ou 3).
- 3° Si n est pair, on connaît le vainqueur à l'issue de la 2° étape et nous avons 2 compétiteurs pour la 2° place, que nous départageons par une comparaison. Si n est impair, notons par $n = 2k + 1$ le joueur laissé de côté; notons par $2k$ le vainqueur de la 2° étape, et $2k - 1$ (resp. $2k - 2$) le compétiteur pour la 2° place qui a perdu devant $2k$ à l'étape n° 1 (resp. n° 2).

Dans la figure 1, ci-dessous, la branche de gauche (resp. droite) sous le libellé a vs. b indique que a a perdu devant b (resp. a a gagné devant b); le sommet pendant de l'arborescence E_b^a indique le résultat final suivant: a est vainqueur et b est «sous-vainqueur».

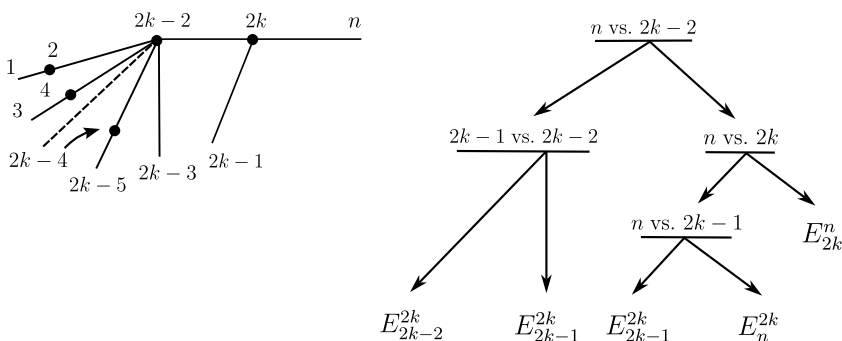


Figure 1

Procédure R_I .

C'est une procédure semi-récurrente où l'on ne fait pas de couplages. Comme ci-dessus écrivons $n = 2k$ ou $2k+1$. Tout d'abord nous partitionnons les n joueurs en 2 sous-ensembles d'au moins k joueurs, sans faire aucune comparaison.

Pour $n \geq 4$, nous devons effectuer alors 3 étapes qui consistent à :

- 1° Par récurrence, chercher le meilleur joueur de chaque sous-ensemble (procédure évidente pour $n = 2$ ou 3) et garder la trace de tous les compétiteurs pour la 2^e place.
- 2° Comparer les deux vainqueurs afin de définir le vainqueur; le perdant devient alors un compétiteur pour la 2^e place, mais pas ceux qu'il a vaincu. Supposons qu'il y ait alors $c \geq 2$ compétiteurs en course pour la 2^e place.
- 3° Effectuer un *tournoi* «K.O.» à $c - 1$ parties pour déterminer le «sous-vainqueur».

Procédure R_M

Nous utilisons, de nouveau, ici, le couplage total et le développement binaire de n ; 3 étapes :

- 1° Trouver le vainqueur de chaque sous-ensemble de 2^{r_i} joueurs ($r_i > 0$).
- 2° Faire jouer le vainqueur du sous-ensemble r_s (c'est-à-dire le plus petit) contre celui du sous-ensemble r_{s-1} (le deuxième plus petit). Faire jouer ce vainqueur contre celui du sous-ensemble r_{s-2} , etc., jusqu'à déterminer le vainqueur de tous. Soit c le nombre de compétiteurs pour la 2^e place.
- 3° Faire un tournoi «K.O.» à $c-1$ parties pour déterminer le «sous-vainqueur».

Procédure R_P

Cette procédure récurrente est due à C. F. Picard [17]; pour $t = 2$, elle est analogue à celle de Steinhaus pour $t = n - 1$. Notons par 1, 2, ..., n les joueurs répartis en ordre aléatoire. L'algorithme itératif de la procédure se déroule en 3 étapes.

- 1° 1 joue contre 2; supposons que 1 ait perdu.
- 2° 3 joue contre le perdant (1); si 3 perd, on l'élimine. Si 3 gagne, on élimine 1; 3 joue alors contre 2, pour ordonner le couple (3, 2).
- 3° Dans l'un ou l'autre cas, nous obtenons une paire ordonnée; on répète le processus ci-dessus pour tous les joueurs en liste.

Bien que cette procédure soit remarquable par sa simplicité et qu'elle soit bien adaptée à l'analyse et au calcul sur machine, nous montrerons plus tard qu'on ne peut pas en tenir compte. Elle est cependant utile en tant qu'élément de comparaison standard pour le problème « E »; on la suppose optimale dans la classe des procédures récurrentes pour $t = 2$.

Procédure R_I

C'est une procédure «semi-récurrente», qui débute par la même étape que R_I ; nous l'omettrons ici. Pour $n = 2$ et 3, nous utilisons les procédures évidentes; supposons $n \geq 4$; les étapes suivantes sont:

- 2° Par récurrence, on détermine sur chaque sous-ensemble le vainqueur et le «sous-vainqueur». Supposons que les 2 couples obtenus soient (a, b) et (c, d) où b et d désignent les deux vainqueurs.
- 3° Pour $n \geq 4$ la 3° étape consiste en deux parties décrites figure 2.

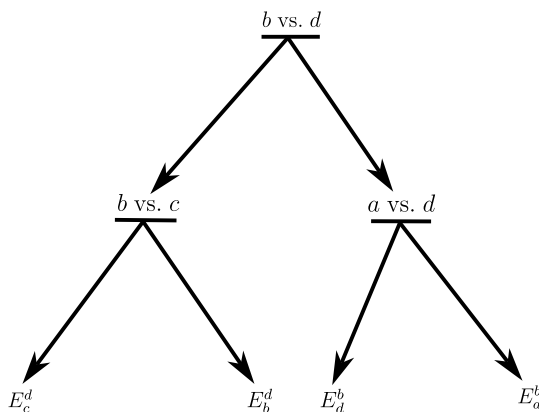


Figure 2

Bien que R_I soit pauvre du point de vue longueur de cheminement, nous l'incluons dans notre tableau de comparaison; elle illustre l'importance de subtiles différences entre procédures.

*Tableau de comparaison entre les 8 procédures
pour $t = 2$*

<i>Valeurs de la longueur de cheminement</i>									
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
BI*	1	2,584	3,917	4,922	5,773	6,488	8,380	9,057	9,668
BIC*	1	2,5	4,0	5,0	6,5	7,5	9,0	10,0	11,0
R_{E_1}	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{15}{30}$	$7\frac{170}{210}$	9	$10\frac{84}{1260}$	$11\frac{112}{3780}$
R_E	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{15}{30}$	$7\frac{170}{210}$	$9\frac{5}{840}$	N.C.	N.C.
R_{E_2}	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{20}{30}$	$7\frac{170}{210}$	9	$10\frac{84}{1260}$	$11\frac{112}{3780}$
R_F	1	$2\frac{2}{3}$	4	$5\frac{8}{30}$	$6\frac{20}{30}$	$7\frac{176}{210}$	9	$10\frac{160}{1260}$	$11\frac{1008}{3780}$
R_{I^*}	1	$2\frac{2}{3}$	4	$5\frac{12}{30}$	$6\frac{20}{30}$	$7\frac{180}{210}$	9	$10\frac{420}{1260}$	$11\frac{1512}{3780}$
R_M	1	$2\frac{2}{3}$	4	$5\frac{18}{30}$	$6\frac{20}{30}$	$7\frac{180}{210}$	9	$10\frac{840}{1260}$	$11\frac{2268}{3780}$
R_P	1	$2\frac{2}{3}$	$4\frac{1}{6}$	$5\frac{17}{30}$	$6\frac{27}{30}$	$8\frac{39}{210}$	$9\frac{366}{840}$	$10\frac{829}{1260}$	$11\frac{3243}{3780}$
R_I	1	$2\frac{2}{3}$	4	$5\frac{20}{30}$	$7\frac{10}{30}$	$8\frac{140}{210}$	10	$11\frac{840}{1260}$	$13\frac{1260}{3780}$
<i>Valeur du minimax</i>									
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
BIS*	1	3	4	6	7	8	9	11	12
R_M	1	3	4	6	7	8	9	11	12
R_{I^*}	1	3	4	6	7	8	9	11	12
R_F	1	3	4	6	7	9	9	11	12
R_{E_2}	1	3	4	6	7	9	9	11	12
R_{E_1}	1	3	4	6	8	9	9	11	12
R_E	1	3	4	6	8	9	11	N.C.	N.C.
R_I	1	3	4	6	8	9	10	12	14
R_P	1	3	5	7	9	11	13	15	17
BI* Borne inférieure pour toutes les procédures utilisant le couplage cyclique. BIC* Borne inférieure conjecturée: $n - 2 + 0,5[2 \cdot \log(n)]$. BIS* Cette borne inférieure pour M est due à Schreier et vaut: $n - 1 + \lfloor \log(n - 1) \rfloor.$ N.C. Non calculé.									

CHAPITRE 3

RELATIONS ET PROPRIÉTÉS

Puisque ce sont les procédures les plus simples qui fournissent les meilleurs résultats, nous verrons ici les procédures dans l'ordre inverse de celui du chapitre précédent.

A) Pour la procédure R_I , notons par $f_6(n)$ la longueur de cheminement pour $t = 2$. De la définition de R_I , nous tirons, pour $m \geq 2$, les formules de récurrence qui suivent:

$$(3.1) \quad \begin{cases} f_6(2m) = 2f_6(m) + 2 \\ f_6(2m+1) = f_6(m) + f_6(m+1) + 2 \end{cases} \quad \text{avec} \quad \begin{cases} f_6(2) = 1 \\ f_6(3) = 2 + \frac{2}{3}. \end{cases}$$

De l'équation (3.1) on obtient par récurrence pour $n = 2m = 2^r$ et $r \geq 1$

$$(3.2) \quad f_6(2^r) = 3 \times 2^{r-1} - 2.$$

Pour $n = 2^r + c$ ($0 \leq c < 2^r$) nous posons:

$$f_6(n) = 3 \times 2^{r-1} - 2 + g_6(c) + k \cdot 2^r \text{ dans (3.1).}$$

En utilisant une condition initiale pour montrer que $k = 0$, on obtient les relations homogènes plus simples:

$$(3.3) \quad \begin{cases} g_6(2c) = 2g_6(c) \\ g_6(2c+1) = g_6(c) + g_6(c+1) \end{cases} \quad \text{avec} \quad g_6(1) = \frac{5}{3}.$$

Par récurrence dans (3.3) on obtient:

$$(3.4) \quad \begin{aligned} g_6(c) &= \frac{5}{3}c \text{ et de là, pour tout } n \geq 2: \\ f_6(n) &= f_6(2^r + c) = 3 \times 2^{r-1} - 2 + \frac{5c}{3}. \end{aligned}$$

Pour la procédure R_I il est curieux de constater que tout hasard peut être déposé jusqu'à $n = 3$.

Soit $\overline{f_6}(n)$ le nombre maximum de comparaisons requises dans R_I pour $t = 2$; les équations pour $\overline{f_6}(n)$ sont les mêmes que dans (3.1), la seule différence est que la deuxième condition initiale devient $\overline{f_6}(3) = 3$.

Comme plus haut, $\overline{g_6}(c) = 2c$ et de là, pour tout $n \geq 2$

$$(3.5) \quad \overline{f_6}(n) = \overline{f_6}(2^r + c) = 3 \times 2^{r-1} - 2 + 2c.$$

Nous verrons plus loin d'autres procédures, qui pour $n \rightarrow \infty$ sont, de la forme $n + \log(n)$. Du fait que $f_6(n) \geq \frac{3n}{2} - 2$, il s'ensuit que R_I n'est pas efficace pour de grandes valeurs de n . En outre, plusieurs autres procédures sont meilleures que R_I pour $n \geq 2$ au sens de la longueur de cheminement. Des remarques semblables peuvent être faites sur la longueur du plus long chemin.

B) Soit $f_5(n) = E\{T|R_P\}$ pour $t = 2$.

Puisque le $j^{\text{ième}}$ joueur ($j \geq 3$) gagne son premier match (et de là joue un match supplémentaire) avec la probabilité $\frac{2}{j}$, il s'ensuit que pour tout $n \geq 2$:

$$(3.6) \quad f_5(n) = n - 1 + \sum_{j=3}^n \frac{2}{j} = n - 4 + 2 \sum_{j=1}^n \frac{1}{j} \approx n + 2 \log_e(n).$$

En clair, si les joueurs $j = 2, 3, \dots, n$, gagnent, nous obtenons la longueur maximum $\overline{f_5}(n)$; d'où pour tout $n \geq 2$;

$$(3.7) \quad \overline{f_5}(n) = 2n - 3.$$

Bien que R_P ait une longueur de cheminement meilleure que R_I , sa valeur de chemin maximum est beaucoup plus mauvaise; ces résultats ont déjà été mis en évidence dans notre tableau pour $n \leq 10$. Pour tout n et pour n tendant vers l'infini, nous avons:

$$(3.8) \quad f_5(n) \leq n - 4 + 2 \left(\log_e(n) + \gamma + \frac{1}{2n} \right),$$

où $\gamma \approx 0,577$ est la constante d'Euler.

On peut utiliser ce résultat pour montrer que $f_5(n)$ est plus petit que $\frac{3n}{2} - 2$ et partant, plus petit que $f_6(n)$ pour tout $n \geq 2$.

C) Soit $f_4(n) = E\{T|R_M\}$ pour $t = 2$.

Soit $n = 2^{r_1} + 2^{r_2} + \dots + 2^{r_s}$ en notation binaire; ceci partitionne les n joueurs au hasard en s sous-ensembles consécutifs de tailles 2^{r_i} ($i = 1, 2, \dots, s$) avec $r_1 > r_2 > \dots > r_s \geq 0$. Dans ces sous-ensembles nous avons besoin d'un total de $n - s$ comparaisons pour trouver les s meilleurs joueurs et entre les s sous-ensembles il nous faut $s - 1$ comparaisons supplémentaires pour trouver le plus fort de tous. Le vainqueur du $j^{\text{ième}}$ sous-ensemble a la probabilité $\frac{2^{r_j}}{n}$ d'être le meilleur. Puisque nous faisons un tournoi «K.O.» à l'intérieur de chaque sous-ensemble et à cause de l'ordre au rang 2, ce vainqueur traîne avec lui r_j prétendants à la 2^e place à l'intérieur de son propre sous-ensemble, $j - 1$ autres venant des $j - 1$ sous-ensembles plus grands et $1 - \delta_{js}$ des sous-ensembles plus petits.

Ici $\delta_{js} = 1$ si $j = s$ et 0 si $j < s$. Ainsi, si le $j^{\text{ième}}$ sous ensemble-produit est le meilleur, il faut $r_j + (j - 1) + (1 - \delta_{js}) - 1$ comparaisons supplémentaires.

De là, pour tout $n \geq 2$

$$(3.9) \quad \begin{aligned} f_4(n) &= (n - s) + (s - 1) + \frac{1}{n} \sum_{j=1}^s (r_j + j - 1 - \delta_{js}) 2^{r_j} \\ &= n - 2 + \frac{1}{n} \sum_{j=1}^s (r_j + j - \delta_{js}) 2^{r_j}. \end{aligned}$$

Il n'est pas facile d'en faire l'étude asymptotique; nous pouvons déterminer une borne inférieure pour $f_4(n)$ et utiliser la valeur maximum comme borne supérieure: on obtient une borne inférieure en ne prenant que le premier terme de la sommation dans (3.9).

On remarque que:

$$r_1 = \lfloor \log(n) \rfloor \text{ et que } r_1 - \delta_{1s} = \lfloor \log(n - 1) \rfloor.$$

D'où pour tout $n \geq 2$:

$$(3.10) \quad f_4(n) \geq n - 2 + \frac{2^{\lfloor \log(n) \rfloor}}{n} (1 + \lfloor \log(n-1) \rfloor).$$

Ceci permet de montrer que, pour toute suite de n_i valeurs

$$(3.11) \quad f_4(n_i) \geq n_i - 2 + \frac{1}{2} \log(n_i - 1)$$

et donne une borne inférieure pour la forme asymptotique de $f_4(n_i)$ quand $n_i \rightarrow \infty$.

Le maximum $\overline{f_4}(n)$ requis dans R_M apparaît quand le vainqueur du 1^{er} sous-ensemble (de taille 2^{r_1}) est le vainqueur global. On en déduit:

$$(3.12) \quad f_4(n) \leq \overline{f_4}(n) = (n - s) + (s - 1) + r_1 - \delta_{1s} = n - 1 + \lfloor \log(n - 1) \rfloor.$$

Étant donné que Schreier [19] et Słupecki [20] ont montré que cette valeur était une borne inférieure pour la valeur du minimum de n'importe quelle procédure, il s'ensuit que R_M est une procédure M -optimale.

La procédure R_M est importante aussi, parce qu'elle s'attache à traiter le cas $t = 2$ en séparant le problème de trouver le vainqueur (en tenant compte de l'information supplémentaire recueillie) de celui de déterminer le «sous-vainqueur».

Bien que cette idée ait été déjà utilisée par C. F. Picard (Paragraphe 7.3.1 de la théorie des questionnaires), on pourra remarquer que notre procédure diffère de la sienne. Soit R_{P_1} cette dernière. Il est relativement aisé de montrer que pour tout $n \geq 2$ la procédure R_{P_1} a comme espérance:

$$(3.12a) \quad E\{T|R_{P_1}\} = n - 1 + \frac{2}{n}(n - 2) + \frac{1}{n} \sum_{j=1}^{n-2} (j - 1) = \frac{3}{2}(n - 1) - \frac{1}{n} \approx \frac{3n}{2}$$

qu'il faut comparer à la borne supérieure $n + \log(n)$ obtenue pour R_M dans (3.12). Nous devons dire que R_{P_1} n'est pas admissible des deux points de vue « E » et « M » puisque R_M est au moins aussi bonne pour tout n et, en fait, nettement meilleure pour $n \geq 4$. En particulier dans l'exemple où $n = 5$ envisagé par C. F. Picard R_{P_1} donne 5,8 pour espérance et 7 pour valeur du maximum alors que R_M donne respectivement comme valeurs 5,6 et 6.

D) Posons $f_3(n) = E\{T|R_I\}$ pour $t = 2$. Posons $n = 2^r + c$ et $0 \leq c < 2^r$.

Soit $g_r(n)$ la probabilité pour qu'il y ait r compétiteurs pour la 2^e place après l'étape n° 2 de la procédure R_I . On veut montrer que

$$g_r(n) + g_{r+1}(n) = 1.$$

Supposons, pour cela, que pour tout $n' < 2^r$ (c'est-à-dire n' associé à $r' < r$) le nombre de compétiteurs pour la 2^e place est soit r' , soit $r' + 1$ avec la probabilité 1.

Ensuite pour n pair, $n = 2m$, l'étape n° 3 donne:

$$(3.13) \quad \begin{aligned} g_r(2m) &= g_{r-1}^2(m) + g_{r-1}(m) \cdot g_r(m) = g_{r-1}(m) \\ g_{r+1}(2m) &= g_r^2(m) + g_{r-1}(m) \cdot g_r(m) = g_r(m). \end{aligned}$$

En additionnant les deux équations, nous trouvons encore 1.

De même pour n impair, $n = 2m + 1$:

$$\begin{aligned}
 g_r(2m+1) &= g_{r-1}(m) \cdot g_{r-1}(m+1) + g_{r-1}(m) \cdot g_r(m+1) \cdot \frac{m}{2m+1} \\
 &\quad + g_r(m) \cdot g_{r-1}(m+1) \cdot \frac{m+1}{2m+1} \\
 g_{r+1}(2m+1) &= g_r(m) \cdot g_r(m+1) + g_{r-1}(m) \cdot g_r(m+1) \cdot \frac{m+1}{2m+1} \\
 &\quad + g_r(m) \cdot g_{r-1}(m+1) \cdot \frac{m}{2m+1}.
 \end{aligned}
 \tag{3.14}$$

En faisant la somme, nous obtenons:

$$(g_{r-1}(m) + g_r(m)) \cdot (g_{r-1}(m+1) + g_r(m+1)) = 1.$$

Comme $g_1(2) = 1$ et $g_2(2) = 0$ ce résultat doit être valable pour tout $n \geq 2$.

Puisque r est déterminé par m , nous supprimons l'indice r ; de (3.13) et (3.14) nous déduisons le résultat cherché:

$$\begin{aligned}
 g(2m) &= g(m) \\
 g(2m+1) &= \begin{cases} \frac{m \cdot g(m) + (m+1) \cdot g(m+1)}{2m+1} & \text{si } m+1 \text{ n'est pas une puissance de 2} \\ \frac{mg(m)}{2m+1} & \text{si } m+1 \text{ est une puissance de 2} \end{cases}
 \end{aligned}
 \tag{3.15}$$

et $g(2) = 1$ et $g(3) = \frac{1}{3}$.

Pour $2^r \leq n < 2^{r+1}$ et $r \geq 1$, on vérifie aisément que la solution est:

$$g(n) = \frac{2^{r+1} - n}{n}.$$

Sachant qu'il faut exactement $n - 1$ comparaisons pour trouver le meilleur joueur et qu'il faut $r - 1$ ou r de plus, avec les probabilités respectives $g(n)$ et $1 - g(n)$, pour trouver le «sous-vainqueur» la relation (3.16) donne (dans les mêmes conditions):

$$\begin{aligned}
 f_3(n) &= n - 1 + (r - 1) \cdot g(n) + r \cdot (1 - g(n)) \\
 f_3(n) &= n + r - \frac{2^{r+1}}{n} = n + \lfloor \log(n) \rfloor - \frac{2^{1+\lfloor \log(n) \rfloor}}{n}.
 \end{aligned}
 \tag{3.17}$$

Dans la relation ci-dessus, la plus faible valeur que l'on ajoute à $n - 1$ est $r - 1$ valant $\lfloor \log(n) \rfloor - 1$; la plus grande est $r - \delta_{1s} = \lfloor \log(n - 1) \rfloor$. D'où:

$$n - 2 + \lfloor \log(n) \rfloor \leq f_3(n) \leq n - 1 + \lfloor \log(n - 1) \rfloor.$$

Lorsque n tend vers l'infini $f_3(n)$ tend vers une expression de la forme $n + \log(n)$.

De (3.18), nous tirons la valeur du minimax:

$$\overline{f}_3(n) = n - 1 + \lfloor \log(n - 1) \rfloor.$$

Schreier ayant établi que ce devrait être, au moins là, la valeur du chemin maximum, il s'ensuit que les procédures R_{I^*} et R_M sont toutes deux M -optimales.

E) Posons $f_2(n) = E\{T|R_F\}$ pour $t = 2$. Pour cette procédure nous débutons par le problème du minimax, donc par $\overline{f_2}(n)$. Les détails de la 3^e étape (chap. 2) nous permettent d'écrire, selon la valeur paire ou impaire de n :

$$(3.20) \quad \begin{aligned} \overline{f_2}(2m) &= \overline{f_2}(m) + m + 1 \\ \overline{f_2}(2m + 1) &= \overline{f_2}(m) + m + 3 \end{aligned} \quad \text{où} \quad \begin{cases} \overline{f_2}(2) = 1 \\ \overline{f_2}(3) = 3. \end{cases}$$

En posant $g(n) = \overline{f_2}(n + 1) - \overline{f_2}(n)$ et $\overline{f_2}(1) = -1$ nous obtenons:

$$(3.21) \quad \begin{aligned} g(2m) &= 2 \\ g(2m + 1) &= g(m) - 1 \end{aligned} \quad \text{où } g(1) = 2 \text{ est la seule condition à la limite.}$$

Plus explicitement:

$$(3.22) \quad g(4m + 1) = g(2m) - 1 = 1 = g(3).$$

Si $m = 2c + 1$ est impair, (3.21) et (3.22) donnent:

$$(3.23) \quad g(4m - 1) = g(8c + 3) = g(4c + 1) - 1 = 0 = g(7).$$

De façon générale, si $m = 2^{p-2} \cdot d$ où d est impair et $p \geq 2$, (3.23) donne par itération:

$$(3.24) \quad g(4m - 1) = g(2^p \cdot d - 1) = g(2^{p-1} \cdot d - 1) - 1 = g(4d - 1) - (p - 2) = 2 - p.$$

Il est donc possible maintenant de formuler $\overline{f_2}(n)$ de façon unique pour n pair et n impair en sommant les valeurs $g(j)$ pour $j = 1, 2, 3, \dots, n - 1$ et $\overline{f_2}(1)$. Sans expliciter les calculs, on obtient:

$$(3.25) \quad \overline{f_2}(n) = n - \frac{1 + (-1)^n}{2} + \sum_{j=1}^k (2 - j) \times \left\lfloor \frac{n + 2^j}{2^{j+1}} \right\rfloor \quad \text{et } k = \left\lfloor \log\left(\frac{n}{2}\right) \right\rfloor.$$

Du fait de l'apparition de $(\log(n))^2$ dans le développement limité, il n'est pas évident de montrer que ceci est de la forme $n + \log(n)$, quand $n \rightarrow \infty$.

Cependant pour $n = 2^r$, il est facile de montrer (sans préciser tous les intermédiaires) que $\overline{f_2}(2^r) = 2^r + r - 2$; d'après les résultats de Schreier, pour $n = 2^r$, il résulte qu'il faut au moins:

$$(3.26) \quad n - 1 + \lfloor \log(n - 1) \rfloor = 2^r - 1 + \lfloor \log(2^r - 1) \rfloor = 2^r + r - 2 \text{ comparaisons.}$$

De là nous déduisons que la procédure R_F (pour $t = 2$) est M -optimale, lorsque n vaut 2^r .

Le tableau de comparaison, nous montre que la procédure R_F est légèrement inefficace pour n valant $2^3 - 1 = 7$. Ceci augmente pour $n = 15, 31$ et 63 ; il est tout à fait surprenant de trouver que $\overline{f_2}(n)$ n'est pas monotone.

En fait $\overline{f_2}(15) = 19 > \overline{f_2}(16) = 18$.

Ceci signifie que dans un tournoi de 15 joueurs, il vaudrait mieux (au sens du minimax) introduire un 16^e joueur fictif (disons une fée bienfaitante) qui perdrait tout le temps et ne serait jamais sélectionné comme vainqueur ou sous-vainqueur.

Pour $n = 62$, nous pourrions solliciter l'aide de 2 telles fées, car:

$$\overline{f_2}(62) = 69, \quad \overline{f_2}(63) = 71, \quad \overline{f_2}(64) = 68.$$

Ce défaut de monotonie n'apparaît pas dans les autres procédures et nous pouvons avancer que cela n'arrivera pas pour les procédures entropiques. Ce résultat permet de prouver que R_F n'est pas M -optimale pour $t = 2$, ni, comme le montre notre tableau, E -optimale.

Par ailleurs, en ce qui concerne le problème du tri ($t = n - 1$) la procédure analogue est tout à fait performante; on a montré [8], qu'elle était M -optimale pour $n \leq 11$ et pour $n = 20$, 21.

S. Johnson dans une communication personnelle a affirmé que M. Wells avait aussi montré qu'elle était M -optimale pour $n = 12$ (par calculs sur machine).

Pour $t = 2$ (et $t = n - 1$) la procédure R_F est aussi intéressante pour son espérance relativement basse. Afin de trouver une expression exacte de la longueur de cheminement $f_2(n)$, nous nous servons de l'étape n° 3 pour $n = 2k + 1$; nous calculons les probabilités associées à l'arborescence de R_F du chapitre 2 (fig. 1).

Après l'étape n° 2, le nombre total de cas d'égale probabilité est:

$$(3.27) \quad C = \binom{4}{2} \binom{6}{2} \dots \binom{2k-4}{2} (2k-3)(n-2)n = \frac{(2k+1)!}{k(k+1)2^k}.$$

Pour la 1^{er} comparaison (après l'étape n° 2) ils sont divisés en:

$$(3.28) \quad C_1 = \binom{4}{2} \dots \binom{2k-4}{2} (2k-3)(n-3)(n-1) \text{ et } C - C_1$$

pour les branches de gauche et de droite; les probabilités respectives sont alors:

$$(3.29) \quad p_1 = \frac{(n-3)(n-1)}{n \cdot (n-2)} \text{ et } p_2 = \frac{2n-3}{n \cdot (n-2)}.$$

On peut calculer de même les deux probabilités pour le seul embranchement restant dans notre arbre de R_F (fig. 1):

$$(3.30) \quad p_{21} = \frac{n-1}{2n-3} \text{ et } p_{22} = \frac{n-2}{2n-3}.$$

Nous en déduisons ainsi la longueur de cheminement associée à l'étape n° 3 pour n impair:

$$(3.31) \quad 2(p_1 + p_2 \cdot p_{22}) + 3p_2 \cdot p_{21} = 2 + \frac{n-1}{n(n-2)} < 3,$$

au lieu du 3 utilisé dans la 2^e équation de (3.20). Nous devons ainsi soustraire $1 - \frac{n_i-1}{n_i(n_i-2)}$ pour chaque entier impair $n_i \geq 3$ apparaissant dans la suite $n, \lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{4} \rfloor$; supposons qu'il y ait t entiers de la sorte: n_1, n_2, \dots, n_t . Alors nous obtenons:

$$(3.32) \quad f_2(n) = \overline{f_2}(n) - t + \sum_{i=1}^t \frac{n_i-1}{n_i(n_i-2)} \text{ ou } \overline{f_2}(n) \text{ est donné par (3.25).}$$

Bien que nous ne prouvions pas que $f_2(n)$ soit strictement croissante avec n , le tableau du chapitre 2 montre que ce n'est pas faux et le calcul le montre pour $n = 15, 16, 62, 63$ et 64 . En particulier, pour $n = 2^r$, la relation (3.32) nous donne:

$$(3.33) \quad f_2(2^r) = \overline{f_2}(2^r) = 2^r + r - 2.$$

On n'a pas prouvé que $2^r + r - 2$ était une borne inférieure de $E\{T|n = 2^r\}$ pour toutes les procédures, mais on peut supposer que c'est vrai. On peut montrer assez aisément que cette borne inférieure est valable pour toutes les procédures d'une certaine classe (classe où l'on a sélectionné le vainqueur par les $n - 1$ premières comparaisons) mais le résultat général n'est pas encore certain.

On peut montrer que la procédure R_F est la meilleure, chaque fois que l'on doit utiliser les deux premières étapes de R_F , à savoir le couplage ordinaire et une procédure semi-récurrente sur les vainqueurs; Hadian dans [10] tient compte de ces résultats.

F) Posons $f_1(n) = E\{T|R_E\}$, pour $t = 2$. Pour les procédures entropiques, nous n'avons pas de formules exactes pour tout n , et partant, les résultats sont moins complets. Les résultats numériques constituent la majorité de l'information sur la valeur performante de telles procédures. Nous allons décrire la procédure R_E , en détail, pour $n = 6$. Le tableau du chapitre 2 montre, que pour $n \leq 10$, les meilleurs résultats sont obtenus par l'une des 3 procédures R_E ; R_{E_1} apparaît en particulier être la meilleure de toutes.

Sans le secours de formules exactes, nous ne pouvons montrer que la longueur de cheminement $f_1(n)$ de R_E est de la forme $n + \log(n)$, quand $n \rightarrow \infty$, comme pour R_I ; on peut supposer que c'est vrai. Dans le prochain chapitre, nous en déduisons les bornes inférieures de l'espérance de R_{E_1} et R_{E_2} . Dans le tableau, nous donnons comme borne inférieure de toutes les procédures où $t = 2$, la valeur $n - 2 + 0,5[2 \log(n)]$, supposée vraie.

Voici maintenant, pour $n = 6$, les détails des calculs d'une étape de R_E .

Nous avons vu auparavant que la procédure compare 1 à 2, puis (pour tout $n \geq 4$) 3 à 4 et enfin, supposant que les vainqueurs soient les nombres pairs, 2 à 4. Après cela une procédure à «couplage total», définie plus loin, compare 5 à 6 et comme le montrera le prochain chapitre, ceci réduit l'entropie de $\frac{2(2n-3)}{n(n-1)}$ (soit de $\frac{3}{5}$ pour $n = 6$).

Nous désirons montrer que R_E n'utilise pas cette comparaison, car la comparaison 5 à 2 procure une réduction d'entropie plus importante.

L'espérance de l'incertitude: $E\{I\}$. Après la comparaison de 2 à 4 (supposant 4 et 2 les vainqueurs précédents), en utilisant (4.7) ou (4.8) ou un calcul direct on montre que cette espérance vaut:

$$(3.34) \quad E\{I\} = \log(30) - \frac{32}{15} = 2,773...$$

La probabilité pour que 2 batte 5 (resp. 5 batte 2) après que 4 ait battu 2, est $\frac{8}{15}$ (resp. $\frac{7}{15}$) ce que l'on peut calculer aisément.

Si 5 perd contre 2, les différents cas possibles restants sont:

- 1 sous-ensemble, noté D_2^4 , de 24 cas
- 3 sous-ensembles, notés D_3^4 , D_4^6 , D_6^4 , ayant 8 cas chacun.

Pour la branche de gauche, nous avons 48 cas.

Si 5 gagne 2, les différents cas possibles restants sont:

- 2 sous-ensembles, notés D_4^5 et D_5^4 , de 12 cas chacun
- 3 sous-ensembles, notés D_3^4 , D_4^6 , D_6^4 , de 4 cas chacun
- 2 sous-ensembles, notés D_5^6 et D_6^5 de 3 cas chacun.

Ce qui fait un total de 42 cas pour cette branche de droite. Nous avons noté ici par D_i^j , l'issue possible: j est le vainqueur et i le «sous-vainqueur», d'où nous tirons la valeur de l'espérance de l'incertitude, après la comparaison de 5 à 2:

$$\begin{aligned}
 E\{I\} &= \frac{8}{15} \times \left(\frac{1}{2} \log(2) + \frac{1}{2} \log(6) \right) + \frac{7}{15} \\
 (3.35) \quad &\times \left(\frac{4}{7} \log\left(\frac{7}{2}\right) + \frac{2}{7} \log\left(\frac{21}{2}\right) + \frac{1}{7} \log(14) \right) \\
 &= \frac{1}{5} + \frac{2}{5} \log(3) + \frac{7}{15} \log(7) \approx 2,144.
 \end{aligned}$$

La réduction d'entropie vaut $(\log(30) - \frac{32}{15}) - (\frac{1}{5} + \frac{2}{5} \log(3) + \frac{7}{15} \log(7)) \approx 0,629$, ce qui est plus grand que la réduction obtenue par la comparaison 5 à 6.

Pour R_E , l'arborescence obtenue pour $n = 6$ est:

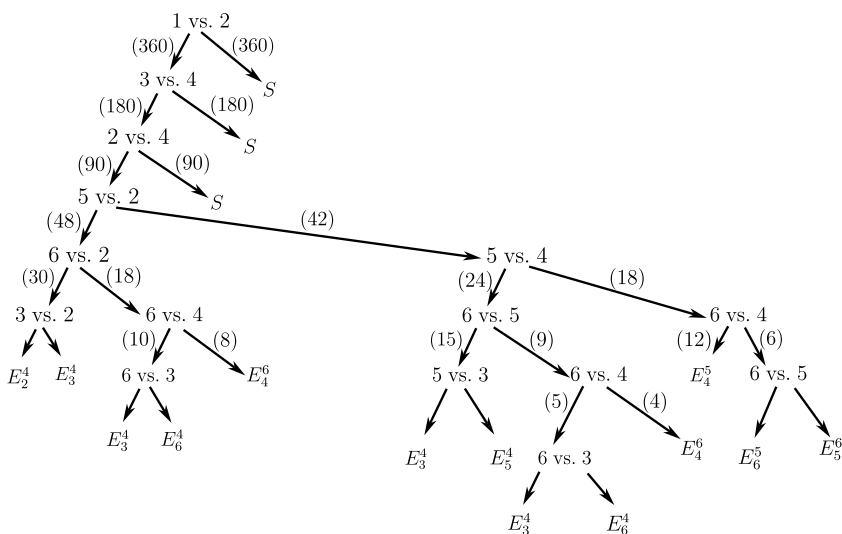


Figure 3

«S» ou «même» signifie ici que la branche droite est la même que la branche gauche correspondante. Les nombres entre parenthèses montrent la répartition des $6! = 720$ cas initiaux (ou états de la nature); ils servent à calculer la longueur de cheminement. Le symbole E_i^j note un sommet pendant de l'arborescence où l'issue est D_i^j : j est le vainqueur et i le «sous-vainqueur».

— Nous n'avons pas trouvé, pour $n = 6$ de procédure ayant une espérance plus faible, mais trois de nos procédures ont un plus long chemin égal à 7.

— Pour $n \leq 4$, les procédures entropiques, comme 3 des autres procédures, sont optimales.

— Pour $n = 5$, elles ont même valeur que la procédure R_F : l'espérance vaut $5\frac{4}{15}$ et la longueur du plus long chemin vaut 6.

— Pour $n = 7$, nous utilisons le couplage total, à savoir les comparaisons successives de: 1 à 2, 3 à 4, 2 à 4 et 5 à 6, étant convenu que le nombre de plus grande puissance de 2 est le vainqueur.

La fin de R_E (ainsi que de R_{E_1} et R_{E_2}) est donnée par:

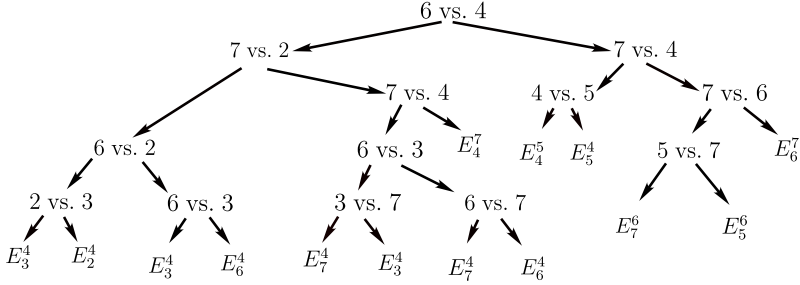


Figure 4

- Valeurs de la longueur de cheminement: $4 + 3\frac{17}{21} = 7\frac{17}{21}$.
- Longueur du plus long chemin: $4 + 5 = 9$.

Il est intéressant de remarquer que R_E compare 6 à 4 (après 1 à 2, 3 à 4, 2 à 4 et 5 à 6) pour tout $n \geq 7$, alors que R_{E_1} et R_{E_2} font la séquence: 7 à 8, puis 6 à 8, puis 4 à 8 pour tout $n \geq 8$. La procédure R_E diffère à la fois de R_{E_1} et R_{E_2} et est plus difficile à mettre en œuvre.

Pour $n = 8$, après la séquence, 1 à 2, 3 à 4, 2 à 4 et 5 à 6 la fin de la procédure R_E est:

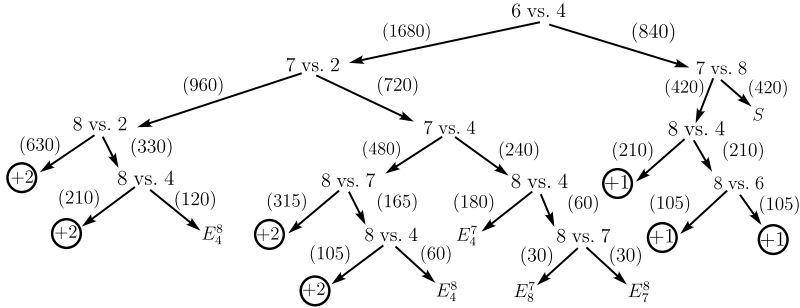


Figure 5

Le symbole $+j$ signifie ici qu'il reste encore j comparaisons évidentes pour finir la procédure. Dans cet exemple on peut calculer rapidement la longueur de cheminement et la longueur du plus long chemin: $9\frac{1}{168}$ et 11.

Pour $n = 8$, la procédure R_{E_1} (identique à R_{E_2} pour $n = 2^r$ quel que soit l'entier r) n'exige que 9 comparaisons en moyenne et la longueur de son plus long chemin vaut 10.

Pour toutes valeurs de n supérieures, on peut avancer que R_{E_1} sera au moins aussi bonne que R_E .

CHAPITRE 4

COUPLAGE CYCLIQUE, COUPLAGE TOTAL,
COUPLAGE ORDINAIRE

Le couplage ordinaire signifie que pour $n = 2k$ ou $n = 2k + 1$, on ne fait que k comparaisons; lorsque n vaut 2^r , un tournoi «K.O.» pour déterminer le vainqueur, consiste à effectuer un couplage ordinaire de tous les vainqueurs du tour précédent. Ainsi le nombre de tours est r et le nombre total de comparaisons est $n - 1$.

Pour définir couplage cyclique et couplage complet (ou total), on utilisera le lemme suivant:

Lemme 1: Soit p la plus grande puissance de 2 que l'on peut mettre en facteur dans $n!$. Écrivons

$$n! = 2^p(2c + 1), \quad c \geq 0 \text{ et entier.}$$

Soit s défini par:

$$(4.1) \quad n = 2^{r_1} + 2^{r_2} + 2^{r_3} + \dots + 2^{r_s} \text{ où } r_1 > r_2 > r_3 > \dots > r_s \geq 0.$$

Alors

$$(4.2) \quad p = n - s = \sum_{i=1}^s (2^{r_i} - 1) = \sum_{j=1}^{r_1} \left\lfloor \frac{n}{2^j} \right\rfloor.$$

Démonstration par récurrence sur n

— Pour n pair, le passage de n à $n + 1$ est évident car p n'a pas changé et s croît de 1 lorsque n augmente de 1.

— Pour n impair, $r_s = 0$. Supposons que $n + 1$ remplace

$$1 + 2 + \dots + 2^{j-1} \text{ par } 2^j.$$

p s'accroît alors de la quantité j et s décroît de $j - 1$. Ce résultat est valable pour $n = 1$: le lemme est démontré.

La dernière égalité de la relation (4.2) se démontre aussi par récurrence en considérant les deux cas n pair et n impair.

Dans (4.2), la première sommation nous montre que p est exactement le nombre de comparaisons nécessaires pour définir le vainqueur dans chacun des sous-ensembles de taille 2^{r_i} , $i = 1, 2, 3, \dots, s$.

— Pour le *couplage total*, on définit les s sous-ensembles de la relation (4.1) et faisons les p comparaisons nécessaires pour trouver le meilleur joueur de chaque sous-ensemble; on a utilisé ce type de couplage dans R_{E_2} et R_M .

— Pour le *couplage cyclique*, on ne fait que les $2^{r_1} - 1$ comparaisons nécessaires pour trouver le vainqueur du plus grand sous-ensemble de taille 2^{r_1} ainsi qu'il est défini en (4.1). Pour $n = 2^r$ ces deux types sont évidemment les mêmes.

Parmi les procédures E -optimales, nous supposons qu'il y a une procédure à couplage cyclique; il est intéressant de voir quelles sont les propriétés d'une telle procédure (nous la noterons R_C). C'est l'objet de cette 4^e partie.

Lorsque $n = 2^r$, les procédures à couplage cyclique (comme celles à couplage total) nécessitent $n - 1$ comparaisons pour nous fournir le vainqueur et r compétiteurs pour la 2^e place. Par ailleurs, il nous faut encore exactement $r - 1$ comparaisons supplémentaires pour déterminer le sous-vainqueur. Donc pour toute procédure R_C , avec $n = 2^r$ nous avons:

$$(4.3) \quad E\{T|R_C\} = 2^r + r - 2 = \max\{T|R_C\}.$$

Nous obtenons ainsi une borne inférieure pour chaque type de couplage. Si parmi les procédures E -optimales, il y avait une procédure à couplage cyclique, la borne inférieure pour tout R_C devrait aussi être borne inférieure pour toutes les procédures.

Réduction d'entropie

On dit qu'une comparaison C_j (ou C_j (a vs. b)) est de *degré* j si les 2 joueurs a et b ont $2^{j-1} - 1$ joueurs connus de force moindre, ces deux ensembles étant disjoints, et si aucun des deux joueurs n'a rencontré un joueur de force supérieure ($j = 1, 2, 3, \dots, \lfloor \log(n) \rfloor$).

Nous voulons démontrer une propriété de la réduction d'entropie valable pour toute comparaison de degré j , sans nous soucier de savoir où cela apparaît dans nos procédures.

Tout d'abord pour $j = 1$. Considérons C_1 (a vs. b) et supposons que nous ne puissions pas connaître les résultats des comparaisons sur les $n - 2$ joueurs restants.

Lemme 2: La réduction d'entropie r_1 (a vs. b) due à la comparaison du 1^{er} degré C_1 est donnée par

$$(4.4) \quad r_1 (a \text{ vs. } b) = \frac{2(2n - 3)}{n(n - 1)},$$

indépendamment de ce que l'on savait déjà sur l'ordre concernant les $n - 2$ joueurs restants.

Démonstration

On prend par commodité $a = 2k + 1$ et $b = 2k + 2 \leq n$; on suppose que tout ce que l'on connaît ne concerne que les joueurs $1, 2, 3, \dots, 2k$.

Considérons un ordre défini sur ces $2k$ joueurs, noté par exemple: $1 < 2 < 3 < \dots < 2k$, où $<$ signifie inférieur à.

On écrit alors la partition en sous-ensembles restants des cas correspondant aux issues possibles D_i^j ($i, j \geq 2k$), D_i^{2k} , D_{2k}^i ($i > 2k$) et D_{2k-1}^{2k} ; on obtient le nombre de cas ou probabilité relative pour chacun avant d'avoir fait la comparaison C_1 :

$$(4.5) \quad \begin{array}{ll} (n - 2k)(n - 2k - 1) & \text{sous-ensembles de } \frac{(n-2)!}{(2k)!} \text{ cas chacun,} \\ 2(n - 2k) & \text{sous-ensembles de } \frac{(n-2)!}{(2k-1)!} \text{ cas chacun,} \\ 1 & \text{sous-ensemble de } \frac{(n-2)!}{(2k-2)!} \text{ cas.} \end{array}$$

La probabilité de chaque sous-ensemble est calculée comme le rapport de son nombre de cas sur le nombre total de cas possibles; on trouve $\frac{n!}{(2k)!}$ cas possibles au total.

L'incertitude $E_1\{I\}$, avant de faire la comparaison C_1 , vaut donc:

$$(4.6) \quad \begin{aligned} E_1\{I\} &= \frac{(n-2k)(n-2k-1)}{n(n-1)} \log(n(n-1)) + \frac{(2n-4k)2k}{n(n-1)} \log\left(\frac{n(n-1)}{2k}\right) \\ &\quad + \frac{2k(2k-1)}{n(n-1)} \log\left(\frac{n(n-1)}{2k(2k-1)}\right) \\ E_1\{I\} &= \log(n(n-1)) - \frac{2k(2n-2k-1)}{n(n-1)} \log(2k) \\ &\quad - \frac{2k(2k-1)}{n(n-1)} \log(2k-1). \end{aligned}$$

En supposant que $2k+2$ batte $2k+1$ (selon notre convention) après avoir fait la comparaison C_1 les deux sous-ensembles correspondants aux deux premières lignes de (4.5) doivent être traités séparément car $2k+1$ est soit mis en première, soit en deuxième place. Nous obtenons donc les 5 cas suivants:

$$(4.7) \quad \begin{array}{ll} 2n-4k-3 & \text{sous-ensembles de } \frac{(n-2)!}{(2k)!} \text{ cas chacun,} \\ (n-2k-2)(n-2k-3) & \text{sous-ensembles de } \frac{(n-2)!}{2(2k)!} \text{ cas chacun,} \\ 2 & \text{sous-ensembles de } \frac{(n-2)!}{(2k-1)!} \text{ cas chacun,} \\ 2n-4k-4 & \text{sous-ensembles de } \frac{(n-2)!}{2(2k-1)!} \text{ cas chacun,} \\ 1 & \text{sous-ensemble de } \frac{(n-2)!}{2(2k-2)!} \text{ cas.} \end{array}$$

Le nombre total de cas possibles est donné par (4.7) égal à $\frac{n!}{2(2k)!}$.

Si d'autre part $2k+1$ bat $2k+2$, on engendre un ensemble symétrique de résultats; on peut alors obtenir de (4.7) l'espérance de l'incertitude $E_2\{I\}$, après la comparaison C_1 .

Comme dans (3.35), en soustrayant (4.6) on obtient le résultat cherché:

$$(4.8) \quad E_2\{I\} - E_1\{I\} = \frac{2(2n-3)}{n(n-1)}.$$

En faisant la moyenne des différentes valeurs du résultat pour diverses valeurs possibles de l'ensemble $\{1, 2, 3, \dots, 2k\}$, on obtient le même résultat (4.7) qu'en connaissant partiellement un sous-ensemble; ceci démontre le résultat.

Un calcul semblable pour toute comparaison C_j de degré j ($j = 1, 2, \dots, [\log(n)]$) donne un résultat plus général:

$$(4.9) \quad r_j(a \text{ vs. } b) = \frac{2^j(2n-1-2^j)}{n(n-1)}.$$

Pour $j > 1$, la démonstration de (4.9) est absolument identique à celle du terme ci-dessus.

Pour $n = 2^r + c$ ($0 \leq c < 2^r$), toute procédure R_C à couplage cyclique a au moins 2^{r-1} couplages de niveau 1, au moins 2^{r-2} couplages de niveau 2, ...,

au moins un couplage de niveau r . On peut supposer que, parmi les $2^r - 1$ premières comparaisons, elle a exactement ces nombres de couplages. La réduction d'entropie due à ces comparaisons est alors de:

$$(4.10) \quad Q = \sum_{j=1}^r \frac{2^j(2n-1-2^j)2^{r-j}}{n(n-1)} = 2^r \left(\frac{(2n-1)r - 2^{r+1} + 2}{n(n-1)} \right).$$

Posons $T_1 = 2^r - 1$, le nombre de ces comparaisons et T_2 le nombre des restantes de façon que $T = T_1 + T_2$. Au départ l'incertitude totale valant $\log(n(n-1))$, et au maximum, la réduction d'entropie à chaque étape valant 1 (en particulier pour les étapes suivant le premier T_1), nous avons:

$$(4.11) \quad Q + (1 \times E\{T_2\}) \geq \log(n(n-1)).$$

On obtient ainsi la borne inférieure cherchée pour toute procédure R_C à couplage cyclique

$$(4.12) \quad E\{T|R_C\} = 2^r - 1 + E\{T_2\} \geq 2^r - 1 + \log(n(n-1)) - 2^r \left(\frac{(2n-1)r - 2^{r+1} + 2}{n(n-1)} \right).$$

Pour $n = 2^r$ on améliore bien sûr cela en utilisant la relation (4.3).

En remarquant que les $p = n - s$ premières comparaisons consistent en $\lfloor \frac{n}{2^j} \rfloor$ comparaisons C_j de niveau j ($j = 1, 2, \dots, r_1$), en utilisant la relation (4.1), on obtient le résultat correspondant dans le cas du couplage total.

En remplaçant dans (4.11) Q par Q_1 on obtient:

$$(4.13) \quad Q_1 = \sum_{j=1}^{r_1} \frac{2^j(2n-1-2^j)}{n(n-1)} \left\lfloor \frac{n}{2^j} \right\rfloor \leq \frac{(2n-1)r_1 - 2^{r_1+1} + 2}{n-1}.$$

De la même façon, pour toute procédure utilisant le couplage complet (comme R_M), on a:

$$(4.14) \quad E\{T\} \geq n - s + \log(n(n-1)) - \sum_{j=1}^{r_1} \frac{2^j(2n-1-2^j)}{n(n-1)} \left\lfloor \frac{n}{2^j} \right\rfloor.$$

Comme dans (4.13), lorsque $n \rightarrow \infty$, la somme peut être bornée. Ici aussi, pour $n = 2^r$ on améliore en utilisant $2^r + r - 2$ d'après (4.3).

Pour le couplage ordinaire, on se sert de $\lfloor \frac{n}{2} \rfloor$ couplages d'ordre 1 seulement et de la même manière pour toute procédure utilisant le couplage ordinaire (comme R_F), on obtient:

$$(4.15) \quad E\{T\} \geq \log(n(n-1)) + \frac{(n-2)(n-3)}{n(n-1)} \left\lfloor \frac{n}{2} \right\rfloor.$$

Ainsi le couplage ordinaire et le couplage cyclique sont tous les deux inclus dans le couplage total; on en déduit que la borne inférieure dans (4.14) n'est pas inférieure à celles de (4.12) et de (4.15). Cependant en avançant qu'il y a une procédure à couplage cyclique parmi celles qui sont E -optimales, la borne inférieure dans (4.12) présente plus d'intérêt.

Dans le tableau du chapitre 2, elle est donnée sans amélioration pour $n = 2^r$.

Dans (4.12), bien que la borne inférieure tende vers n lorsque $r \rightarrow \infty$, on pourra remarquer que pour $n = 3 \cdot 2^{r-1}$ la limite pour $r \rightarrow \infty$ est seulement $\frac{2}{3}(n + \log(n)) + O(1)$.

CHAPITRE 5

PROCÉDURES POUR LE PROBLÈME DE RANGEMENT
AVEC $t = n - 1$

Nous décrivons plusieurs procédures qui sont toutes nouvelles sauf deux, la procédure R_S due à Steinhaus [23] et la procédure R_F due à Ford et Johnson [8]. Notre principal intérêt a porté sur le *concept de maximum de l'espérance mathématique de la réduction de l'entropie sur g rangs* pour les petites valeurs entières de g . Au chapitre 6 nous montrons que pour $g = 1$ ce maximum est obtenu en choisissant la comparaison qui sépare l'ensemble des cas possibles (états de la nature) en deux sous-ensembles de taille égale (autant que possible). Dans la procédure entropique à g rangs, nous espérons obtenir $G = 2^g$ sous-ensembles de taille égale afin de maximiser l'expression:

$$-(p_1 \log(p_1) + p_2 \log(p_2) + \dots + p_G \log(p_G))$$

avec $p_i = \frac{C_i}{T}$, C_i ($i = 1, \dots, G$) nombre de cas du i° sous-ensemble, $T = C_1 + \dots + C_G$ nombre total de cas.

Le concept de couplage complet (exposé aux chapitres 2 et 4) est utilisé pour toutes les nouvelles procédures. Dans les chapitres suivants le terme de longueur de cheminement ne sera plus cité en référence des procédures entropiques.

La procédure R_N utilise le procédé qui consiste à insérer *des items* dans une chaîne principale; on change *l'item* que l'on voulait insérer quand il est évident que l'on introduit un «*bruit*» dans la procédure. Le concept de bruit, le critère pour le détecter, et sa relation à l'espérance seront exposés au chapitre 6.

Procédure R_E : Pour $t = n - 1$, c'est essentiellement une procédure entropique sur un rang, c'est-à-dire, basée sur la recherche de comparaisons binaires qui maximisent l'espérance mathématique de la réduction de l'entropie après une comparaison. En certains points isolés, nous permettons l'utilisation de l'entropie sur 2 ou 3 rangs, sans donner de règle formelle sur ce fait. Un couplage complet est effectué lors des premières comparaisons.

Procédure R_E : Procédure entropique à un rang qui utilise un couplage complet lors des p premières comparaisons. L'entropie est utilisée sur un plus grand nombre de rangs pour décider du choix entre deux comparaisons uniquement quand la réduction de l'entropie sur un rang est égale dans les deux cas.

Procédure R_N : Cette procédure aussi utilise un couplage complet lors des p premières comparaisons; on établit ainsi une chaîne principale (indiquée par les puissances de deux dans notre convention). Ensuite, des items sont insérés dans la chaîne principale, c'est-à-dire, nous comparons uniquement un item hors de la chaîne principale avec un item sur la chaîne. Nous tentons d'insérer l'item choisi jusqu'au moment où il est évident qu'un bruit (indiqué par N) entre dans la procédure (un critère est donné au chapitre 6). La décision d'insérer tel item ou de faire telle comparaison, est essentiellement basée sur une entropie à un rang, c'est-à-dire, étant donné un certain état de connaissance, nous sélectionnons la comparaison qui maximise l'espérance mathématique de la décroissance de l'entropie.

Ces procédures font clairement apparaître que des améliorations sont possibles en utilisant l'entropie sur un plus grand nombre de rangs; le montrer demande des calculs supplémentaires et n'a pas été tenté.

Procédure R_G : Cette procédure se base sur un rangement individuel des sous-ensembles obtenus par le couplage complet, et utilise ensuite le critère de l'entropie à un rang pour fusionner ces séquences dont la taille est égale à une puissance de 2. Pour la différencier de R_E dans le cas où $n = 2^r$, nous supposons que les deux moitiés de taille 2^{r-1} sont rangées séparément puis fusionnées.

Le tableau donne les résultats numériques de ces procédures et les compare à ceux des procédures R_S et R_F . Une omission importante a été commise: la procédure optimale de C. Picard [17] pour $n = 6$ et celle de Cesari [4] pour $n = 7$ qui a 3 unités de bruit seulement, et sans bruit pour $n = 9$, $n = 10$; aucune règle n'est donnée pour n quelconque, dans leurs travaux.

La borne inférieure du tableau est définie par l'expression:

$$LB = r + \frac{2c}{n!}$$

où r et c sont définis par l'expression $n! = 2^r + c$ ($0 \leq c < 2^r$); ce résultat provient des travaux de Huffman [12] et a été appliqué à ce problème par Kislitsyn, d'une manière indépendante, et par l'intermédiaire de la théorie des questionnaires, par Picard [17].

La borne inférieure minimax $MLB = 1 + \lfloor \log(n!) \rfloor$ pour $n \geq 3$ du problème de minimax a été utilisée par Ford et Johnson [8] et aussi étudiée par Steinhaus [25].

Comparaison de six procédures de TRI

BORNE INF- (LB) PROCÉDURES	VALEURS MOYENNES								
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
LB	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{118}{315}$	$15\frac{118}{315}$	$18\frac{1574}{2835}$	$21\frac{11966}{14175}$
R_{E^*}	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{121}{315}$	$15\frac{121}{315}$	$18\frac{1592}{2835}$	N.C.
R_N	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{122}{315}$	$15\frac{122}{315}$	$18\frac{1608}{2835}$	N.C.
R_E	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{26}{45}$	$12\frac{123}{315}$	$15\frac{123}{315}$	$18\frac{1624}{2835}$	N.C.
R_F	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$6\frac{14}{15}$	$9\frac{27}{45}$	$12\frac{144}{315}$	$15\frac{144}{315}$	$18\frac{1656}{2835}$	$21\frac{12060}{14175}$
R_G	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$7\frac{1}{15}$	$9\frac{30}{45}$	$12\frac{150}{315}$	$15\frac{177}{315}$	$18\frac{1737}{2835}$	$21\frac{13725}{14175}$
R_S	1	$2\frac{2}{3}$	$4\frac{2}{3}$	$7\frac{1}{15}$	$9\frac{33}{45}$	$12\frac{186}{315}$	$15\frac{186}{315}$	$18\frac{2304}{2835}$	$22\frac{3015}{14175}$
<i>Bruit</i> (multiplié par dénominateur)									
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
R_{E^*}	0	0	0	0	0	3	3	18	N.C.
R_N	0	0	0	0	0	4	4	34	N.C.
R_E	0	0	0	0	0	5	5	50	N.C.
R_F	0	0	0	0	1	26	26	82	94
R_G	0	0	0	2	4	32	59	163	1759
R_S	0	0	0	2	7	68	68	730	5224
<i>Nombres de Comparaisons</i> (min, max)									
	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
MLB	1	3	5	7	10	13	16	19	22
R_{E^*}	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(18, 20)	N.C.
R_N	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(17, 20)	N.C.
R_E	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(9, 10)	(11, 13)	(14, 16)	(18, 20)	N.C.
R_F	(1, 1)	(2, 3)	(4, 5)	(6, 7)	(8, 10)	(10, 13)	(13, 16)	(16, 19)	(19, 22)
R_G	(1, 1)	(2, 3)	(4, 5)	(6, 8)	(8, 11)	(11, 14)	(14, 17)	(17, 20)	(20, 23)
R_S	(1, 1)	(2, 3)	(4, 5)	(6, 8)	(8, 11)	(10, 14)	(13, 17)	(16, 21)	(19, 25)

CHAPITRE 6

PROPRIÉTÉS ET DÉMONSTRATIONS

La procédure «demi» est définie comme la procédure qui choisit toujours la comparaison qui donnera (autant que possible) deux sous-ensembles de cas de taille égale. Soit T le nombre total d'états de la nature possibles à chaque étape, et soient x et $y = T - x$ le décompte du partage résultant d'une comparaison.

Lemme 1: La procédure «demi» et la procédure entropique sur un rang sont équivalentes.

Démonstration: La décroissance de l'entropie est donnée à chaque étape par l'expression:

$$(6.1) \quad \log(T) - \frac{x}{T} \log(x) - \frac{y}{T} \log(y) = -(p_x \log(p_x) + p_y \log(p_y))$$

avec $p_x = \frac{x}{T}$ et $p_y = \frac{y}{T}$.

Le second terme de (6.1) est maximum pour $p_x = p_y$ soit $x = y$ ce qui démontre la proposition.

Ainsi, si nous pouvons toujours séparer les états de la nature en deux sous-ensembles de taille égale, nous aurons la solution optimale. Les difficultés viennent du fait que ce partage n'est pas toujours possible. Mais il n'est pas nécessaire de séparer chaque ensemble en exactement deux moitiés pour obtenir une structure optimale. Ce dernier point est traité ci-après.

Soit $H(T)$ l'espérance mathématique du nombre de comparaisons nécessaires quand le nombre des états possibles de la nature est T .

Soit x la taille du plus petit des sous-ensembles issus d'une comparaison. Supposons que pour tout sous-ensemble nous pouvons choisir x à notre guise. Alors $H(1) = 0$ et pour $T \geq 2$

$$(6.2) \quad H(T) = 1 + \min_{1 \leq x \leq T/2} \left\{ \frac{x}{T} H(x) + \frac{T-x}{T} H(T-x) \right\}.$$

Soit $h(y) = yH(y)$. Alors (6.2) devient plus simplement

$$(6.3) \quad h(T) = T + \min_{1 \leq x \leq T/2} \{h(x) + h(T-x)\}.$$

Définissons r et c par l'expression $T = 2^r + c$ avec $0 \leq c < 2^r$.

On peut démontrer comme dans le lemme 2 de [21] que le minimum de (6.3) est obtenu pour $x = \frac{T}{2}$ et qu'une valeur exacte de $H(T)$ pour $T > 0$ est donnée [17] par l'expression:

$$(6.4) \quad H(T) = r + \frac{2c}{T} = r + \frac{2}{T}(T - 2^r).$$

Le résultat suivant est utile pour chercher les procédures où le bruit introduit est minimum, en particulier il est utilisé dans la procédure R_N . Il correspond au lemme 3 de [21], mais il faut remarquer ce dernier est, à cause des conditions limites, complètement différent.

Lemme 2 ⁽¹⁾: Pour tout $T \geq 2$ le minimum de (6.3) sera obtenu avec un entier y si et seulement si aucune puissance de 2 est comprise entre y et $T - y$.

¹Ce lemme apparaît dans [17] sous la forme de théorème (chap. 2, page 32-33), dans l'édition de 1965; théorème 10-3 dans l'ouvrage de 1972. Picard appelle *bruit propre* (déf. 12-3, op. cit.) ce qui est appelé, ici, bruit.

Démonstration: Soit $h(x, T)$ la somme entre crochets dans (6.3), à cause de sa symétrie par rapport à $\frac{T}{2}$ nous supposons que $x \leq T - x$. Considérons différentes inégalités entre x , $T - x$, et les puissances de 2 qui encadrent la moyenne $\frac{T}{2}$.

Cas 1: $2^{r-1} \leq x \leq T - x \leq 2^r$.

Soit $r(x)$ la valeur de x telle que $r(x) = r(T - x) = r - 1$. Pour vérifier (6.3) on utilise (6.4) et on calcule

$$(6.5) \quad \begin{aligned} T + h(x, T) &= T + (r - 1)x + 2(x - 2^{r-1}) \\ &\quad + (r - 1)(T - x) + 2(T - x - 2^{r-1}) = rT + 2(T - 2^r) = h(T). \end{aligned}$$

Donc le minimum de (6.3) est obtenu pour une telle valeur de x .

Cas 2A: $2^{s-1} \leq x < 2^s$, $2^r < T - x$ avec $1 \leq s \leq r$.

Cas 2B: $2^{s-1} \leq x < 2^s$, $2^r = T - x$ avec $1 \leq s < r$.

En posant $r(x) = s - 1$ et $r(T - x) = r$, et en faisant un calcul analogue au précédent nous obtenons dans les cas 2A et 2B:

$$(6.6) \quad \begin{aligned} T + h(x, T) &= T + x(s - 1) + 2(x - 2^{s-1}) + (T - x)r + 2(T - x - 2^r) \\ &= h(T) + (T - x - 2^r) + (2^s - x)t + 2^s(2^t - t - 1) \\ &> h(T). \end{aligned}$$

Puisque $t = r - s \geq 0$ et donc $2^t - t - 1 \geq 0$. Dans le cas 2A (resp., 2B) l'inégalité stricte vient du fait que $T - x - 2^r > 0$ (resp., $(2^s - x)t > 0$). Donc pour de telles valeurs de x le minimum de (6.3) ne peut pas être atteint.

Cas 3: $2^{s-1} \leq x < 2^s$, $2^{r-1} < T - x < 2^r$ avec $s \leq r - 1$.

Ici $r(x) = s - 1$, $r(T - x) = r - 1$ et $r - s > 0$. Comme ci-dessus nous obtenons le résultat suivant:

$$(6.7) \quad \begin{aligned} T + H(x, T) &= T + x(s - 1) + 2(x - 2^{s-1}) + (T - x)(r - 1) \\ &\quad + 2(T - x - 2^{r-1}) = h(T) + 2^s(2^t - t - 1) + (2^s - x)t > h(T). \end{aligned}$$

Puisque $t = r - s > 0$ et $2^t - t - 1 \geq 0$, le minimum de (6.3) ne peut pas être obtenu pour de telles valeurs de x .

Cas 4: $x = 2^s$, $2^{r-1} < T - x < 2^r$ avec $s \leq r - 2$.

Ici nous avons $r(x) = s$, $r(T - x) = r - 1$ et $t = r - s \geq 2$.

Comme ci-dessus nous obtenons:

$$(6.8) \quad \begin{aligned} T + H(x, T) &= T + s2^s + (T - 2^s)(r - 1) + 2(T - 2^s - 2^{r-1}) \\ &= h(T) + 2^s(2^t - t - 1) > h(T) \end{aligned}$$

puisque $2^t - t - 1 > 0$ pour $t \geq 2$ le minimum de (6.3) n'est pas atteint.

Puisque ces quatre cas épuisent toutes les combinaisons entre x , $T - x$ et les puissances de 2 encadrant la moyenne $\frac{T}{2}$, le lemme est démontré.

Il découle de ce lemme, qu'en sélectionnant une comparaison à chaque étape de la procédure et en observant la taille des deux sous-ensembles résultants (et leur position relative aux puissances de 2 les plus proches), nous pouvons savoir si une comparaison particulière introduit ou non un bruit dans la procédure. C'est justement le critère que nous avons utilisé dans la procédure R_N . On peut remarquer que le lemme 2 est lié au théorème de Sandelius [18] qui n'emploie pas la même approche et n'obtient pas nos résultats subséquents.

Nous nous intéressons aussi au bruit total introduit dans la procédure, en particulier quand une puissance de deux est strictement comprise entre les tailles des

deux sous-ensembles. Dans les cas 2A, 2B, 3 et 4 cela correspond respectivement à $t = 0, 1, 1$ et 2 .

Dans le cas 2A nous ajoutons $T - x - 2^r$ à $h(T)$, et:

$$(6.9) \quad T - x - 2^r < 2^s - x \quad \text{puisque} \quad s = r \text{ et } T < 2^{r+1}.$$

Dans les cas 2B et 3 nous ajoutons $2^s - x$ à $h(T)$ et

$$(6.10) \quad 2^s - x \leq T - x - 2^s \quad \text{puisque} \quad s = r - 1 \text{ et } T \geq 2^r.$$

Dans le cas 4 nous ajoutons $2^s = 2^{r-2}$ à $h(T)$ et

$$(6.11) \quad 2^{r-2} < T - x - 2^{r-1} \quad \text{puisque} \quad x = 2^{r-2} \text{ et } T > 2^{r-1}.$$

A partir de ces résultats nous pouvons démontrer le lemme suivant.

Lemme 3: Le bruit N dû à une comparaison telle qu'une seule puissance de 2 soit strictement comprise entre les tailles des deux sous-ensembles résultants, $T_1 < 2^a < T_2 = T - T_1$, est tout simplement la différence minimum de cette puissance aux tailles des sous-ensembles:

$$(6.12) \quad N = \min(2^a - T_1, T_2 - 2^a).$$

L'apport de ce bruit N à l'espérance mathématique, si nous avons T cas au départ, est alors $\frac{N}{T}$. Si le nombre de cas au départ est plus grand: $D > T$, alors l'apport est multiplié par la probabilité $\frac{T}{D}$ de parvenir à cette partie de l'arborescence. Donc l'apport total à l'espérance mathématique pour cette comparaison quelconque est $\frac{N}{D}$. Ce dernier résultat est un corollaire du lemme 3, mais son importance est telle que nous préférons le citer comme théorème. Soit les sommets «bruyants» d'une arborescence avec les bruits N_1, N_2, \dots, N_w , un sommet bruyant est dit sommet *bruyant* simple si les tailles des deux sous-ensembles résultant de la comparaison sont strictement séparées par une seule puissance de 2. La valeur de l'espérance mathématique, pour une arborescence sans bruit (qui n'a aucun sommet bruyant) avec à l'origine n états possibles de la nature, est $H(n)$. Nous pouvons énoncer le théorème suivant:

Théorème: Pour toute procédure R qui possède seulement des sommets sans bruit ou des sommets bruyants simples, l'espérance mathématique est donnée par l'expression:

$$(6.13) \quad E\{T|R\} = H(n) + \frac{1}{n} \sum_{i=1}^w N_i,$$

$H(n)$ est donné par l'expression (6.4) et les N_i par l'expression (6.12).

Ce résultat permet de prévoir l'espérance mathématique d'une procédure (espérance mathématique de la longueur de l'arborescence) sans la mettre en œuvre. Il est très utile pour détecter l'existence ou la non existence de bruit dans une arborescence. Il est employé dans la plupart de nos calculs du tableau précédent et aussi dans les notes qui accompagnent les procédures décrites ci-après.

L'analyse précédente présente un intérêt général pour nos problèmes et n'est pas seulement associée aux procédures entropiques, par exemple, l'expression (6.4) s'applique aussi à la procédure de Steinhaus R_S . Puisque la procédure de Steinhaus effectue des insertions individuelles sans bruit, il en résulte que $H(i)$ est l'espérance mathématique du nombre de comparaisons nécessaires pour

insérer un item dans une chaîne de longueur i . Il en résulte alors, suivant (6.4) que l'espérance, sous R_S , pour $2^r \leq n < 2^{r+1}$ est:

$$(6.14) \quad E\{T|R_S\} = \sum_{i=2}^n H(i) = r(n+1) + 2(n-2^r) - \sum_{j=2}^n \frac{2^{1+\lfloor \log(j) \rfloor}}{j}.$$

Une expression semblable a été obtenue par Trybula (communication personnelle); les propriétés asymptotiques ont été étudiées par Kislitsyn [14] et Hadian [10].

Pour la procédure R_F définie dans [8] et mise en œuvre par des formules de récurrence différentes pour les valeurs paires ou impaires de n qui impliquent des sommes compliquées, aucune expression détaillée n'était donnée pour l'entier minimax $U(n)$. Pour tout $n \geq 1$ une expression simple de $U(n)$ est:

$$(6.15) \quad U(n) = j\left(n + \frac{1}{2}\right) - \frac{4}{3}(2^j - 1) - \frac{1}{6} \left(\frac{1 + (-1)^{j+1}}{2} \right).$$

Où $j = \lfloor \log(\frac{3n+2}{2}) \rfloor$. Cette forme a l'avantage de donner rapidement une évaluation asymptotique de $U(n)$ pour $n \rightarrow \infty$, soit

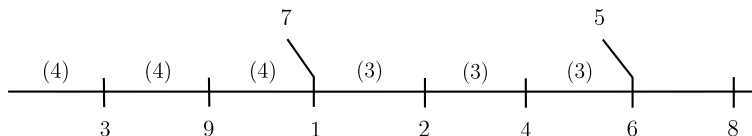
$$(6.16) \quad U(n) = jn - \frac{2^{j+2}}{3} + \frac{1}{2} \log(n) + O(1),$$

où $j = j(n)$ a été défini ci-dessus. Les résultats (6.15) et (6.16) sont dus à Hadian [10].

CHAPITRE 7

REMARQUES AU SUJET DU TABLEAU ET DES ARBORESCENCES

Les arborescences ci-après ne sont qu'un exemple des arborescences construites pour le tableau du chapitre 5. Seules les arborescences les plus embrouillées, avec les résultats les plus originaux, sont montrées. Aucune des arborescences trouvées n'a de meilleurs résultats que la procédure entropique modifiée: R_{E^*} . Cependant on peut raisonnablement espérer que, des procédures utilisant l'entropie sur un plus grand nombre de rangs peuvent améliorer nos résultats. Ceci est basé sur le fait que dans plusieurs cas l'utilisation de l'entropie sur deux rangs améliore franchement le résultat par rapport à l'utilisation de l'entropie sur un rang. Nous en donnons une illustration pour R_E avec $n = 9$. Après 7 comparaisons, un des sommets de l'arborescence est associé à 21 états possibles qui sont représentés par le schéma suivant:



Les obliques indiquent que 5 se place quelque part avant 6, et 7 quelque part avant 1. Si nous insérons 5 le premier, il y a six emplacements où le ranger, et le nombre de cas (ou probabilité relative) pour chacun est donné par le nombre

entre parenthèses. La procédure entropique à un rang nous amène à comparer 5 avec 1 pour obtenir le partage (12, 9) de préférence à 5 avec 9 qui donne le partage (13, 8). Cependant, la procédure entropique sur deux rangs compare le partage en quatre (6, 6, 3, 6) (qui a une unité de bruit) pour le premier coup, avec le partage en quatre (4, 4, 7, 6) pour le dernier coup. Le partage (4, 4, 7, 6) est retenu quand nous utilisons l'entropie sur deux rangs, puisque sa réduction sur deux rangs est alors

$$(7.1) \quad \frac{8}{21} \log\left(\frac{21}{4}\right) + \frac{7}{21} \log\left(\frac{21}{7}\right) + \frac{6}{21} \log\left(\frac{21}{6}\right) \approx 1,957$$

supérieure à 1,952 pour le partage (6, 6, 3, 6).

La procédure R_N est une tentative d'utilisation des résultats précédents, au sujet du bruit, dans la construction des procédures, et le résultat obtenu est satisfaisant. En fait, si la procédure R_N apparaît meilleure que la procédure entropique à un rang R_E , elle n'est pas aussi bonne que la procédure entropique modifiée R_{E^*} .

Le symbole S utilisé dans les arborescences suivantes indique une branche symétrique ou équivalente à une autre branche développée sur sa gauche. Le symbole H , accompagné de l'entier j , sur un arc extrémal, signifie qu'à partir de ce point les rangs suivants correspondent à des insertions sans bruit, l'espérance mathématique de la longueur de l'arborescence correspondant à ces rangs est $H(j)$ (en comptant le sommet figurant sur le schéma). Le symbole H_1 indique que les étapes suivantes ne sont pas des insertions, mais sont évidentes, et sans bruit, telles que le même résultat (6.4) s'applique; on peut considérer que H et H_1 indiquent des choses équivalentes. Les entiers dans les ronds placés entre les arcs partant d'un sommet bruyant indiquent le nombre d'unités de bruit de ce sommet.

Il apparaît qu'aucun bruit ne se produit aux sommets qui correspondent à un nombre de cas inférieur ou égal à 8, mais ceci n'a pas été prouvé.

Aucune des procédures utilisées ne contient un sommet bruyant qui ne soit simple.

Chacune des arborescences débute après les p couplages du couplage complet, p étant l'exposant le plus grand de 2 dans $n!$. Donc le nombre total de cas (états de la nature) est $D = \frac{n!}{2^p}$; qui est le dénominateur commun du tableau du chapitre 5.

Puisque le nombre d'unités de bruit $NU = 3$, avec R_{E^*} et $n = 7$, l'espérance mathématique est (fig. 6):

$$4 + H(D) + \frac{3}{D} = 12 \frac{121}{315} \approx 12,384.$$

Pour $n = 8$ la procédure R_{E^*} est exactement la même sauf trois couplages nouveaux (7 vs. 8, 6 vs. 8, et 4 vs. 8). Donc avec R_{E^*} et $n = 8$ l'espérance mathématique est:

$$15 \frac{121}{315} \approx 15,384.$$

On suppose que ce sont les meilleurs résultats pour $n = 7$ et 8, mais cela n'a pas encore été prouvé. Cesari [4] a montré qu'il n'existe pas de procédure sans bruit pour $n = 7$. A l'aide de nos résultats on pourra essayer de montrer qu'aucune procédure dans ce cas a moins de 3 unités de bruit; ceci n'a pas encore été tenté.

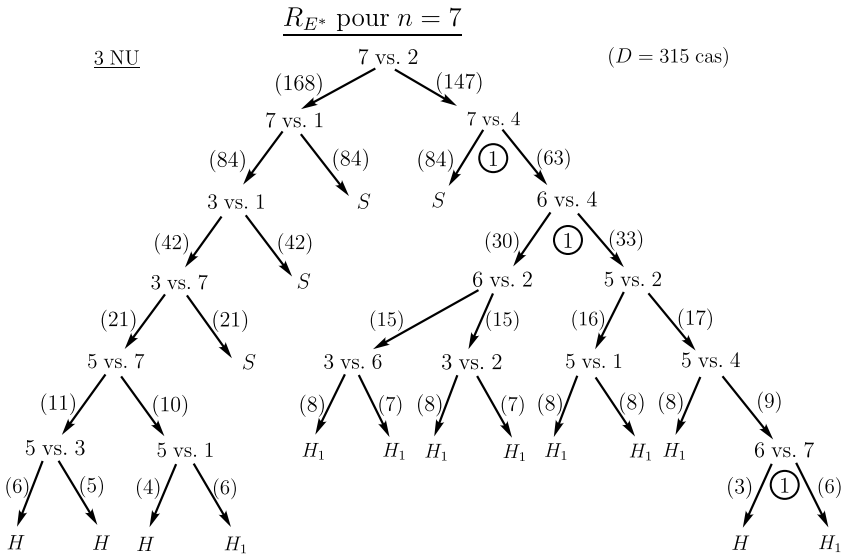


Figure 6

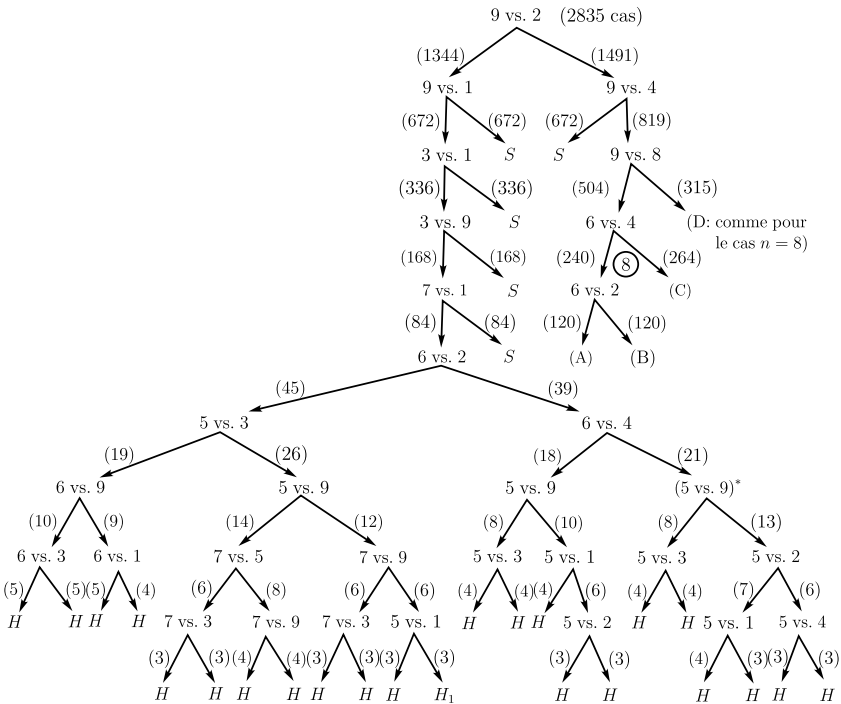
 R_N pour $n = 9$ 

Figure 7

Le bruit total est de 34 unités et donc (fig. 7 à 12)

$$E\{T|R_N\} = 18 \frac{1574 + 34}{D} = 18 \frac{1608}{2835} = 18 \frac{536}{945} = 18,567...$$

L'entropie est utilisée sur deux rangs uniquement aux endroits indiqués par (*) pour éviter une unité de bruit, ou plus précisément 32 unités de bruit à cause des nombreuses branches équivalentes (S).

Le bruit total restant est 18 et donc (fig. 13 à 16):

$$E\{T|R_{E^*}\} = 18 \frac{1574 + 18}{D} = 18 \frac{1592}{2835} = 18,562...$$

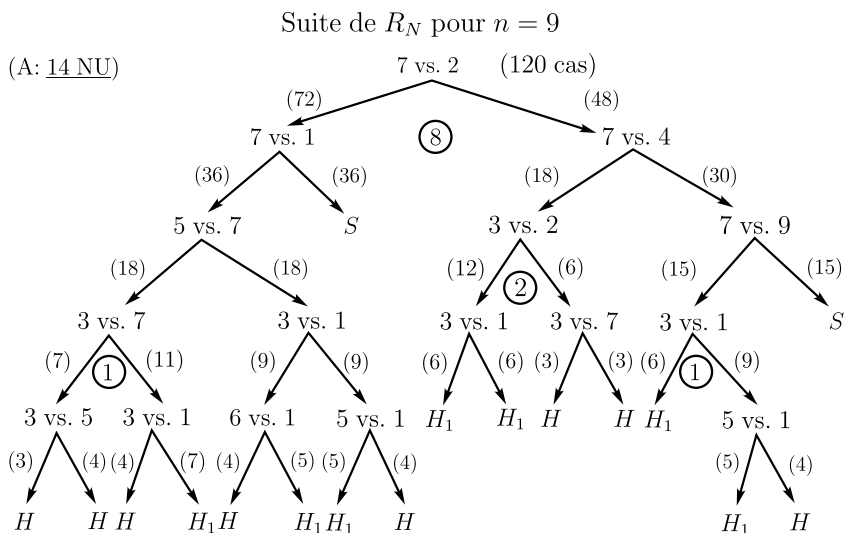


Figure 8

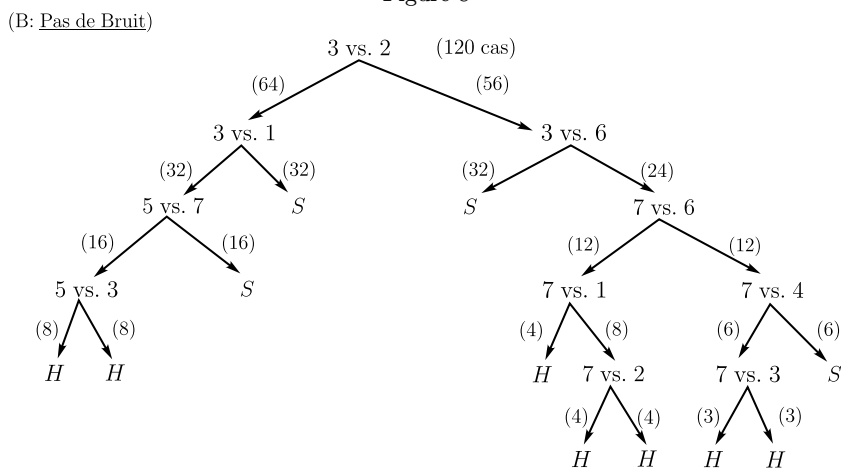


Figure 9

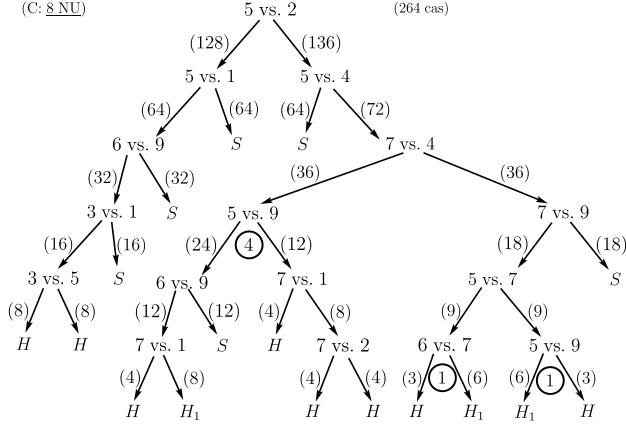


Figure 10

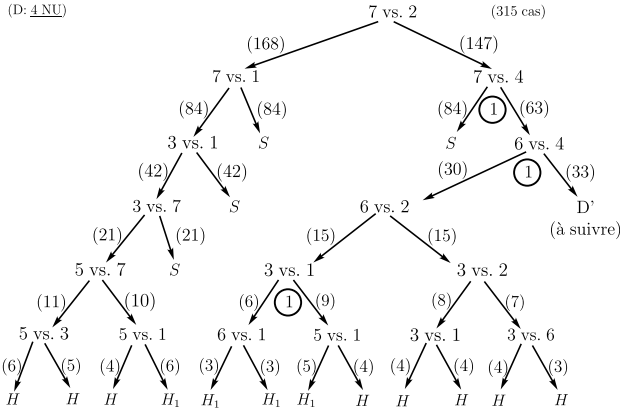


Figure 11

Suite de R_N pour $n = 9$ (D)

(D')

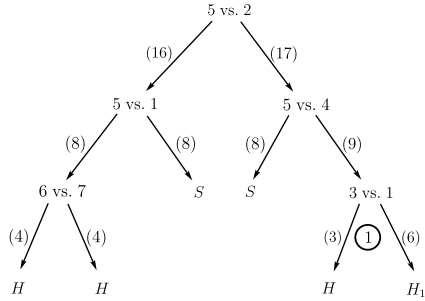


Figure 12

Pour avoir le résultat avec R_E nous prenons $18 + 32 = 50$ NU nous obtenons:

$$E\{T|R_E\} = 18 \frac{1624}{2835} = 18,573...$$

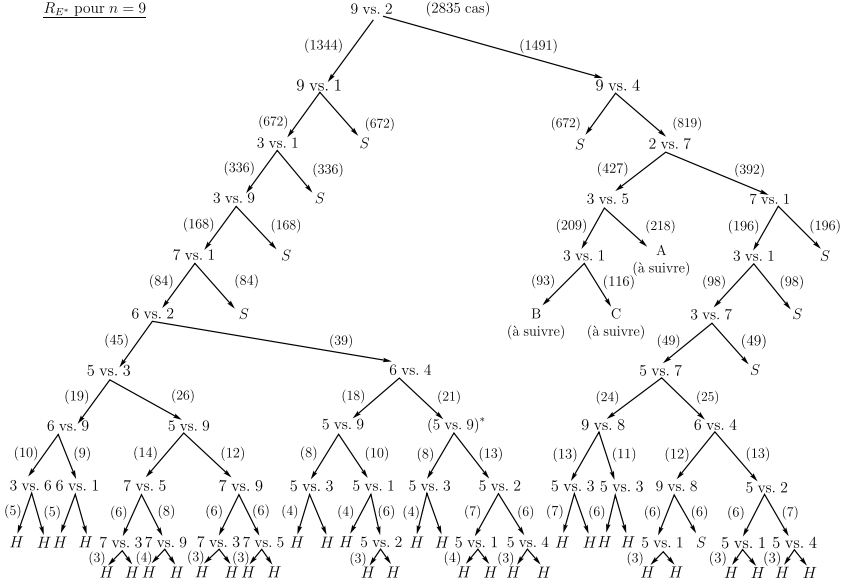


Figure 13

(A)

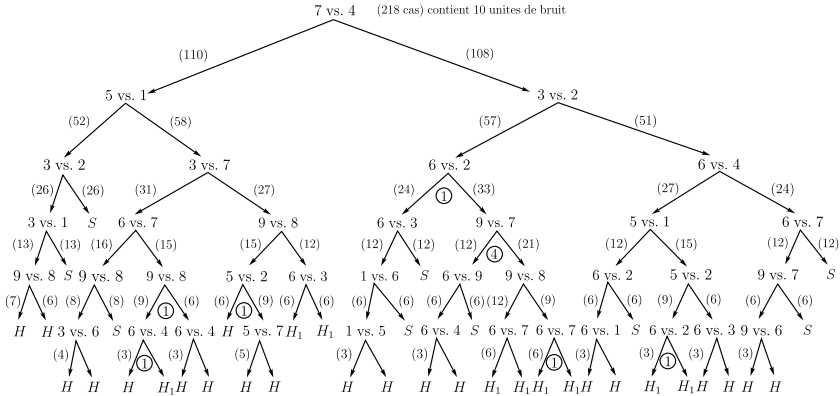
Suite de R_{E^+} pour $n = 9$ 

Figure 14

(B)

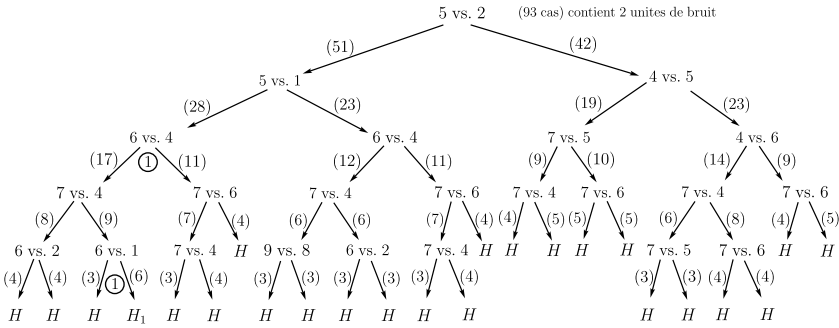
Suite de R_{E^*} pour $n = 9$ 

Figure 15

(C)

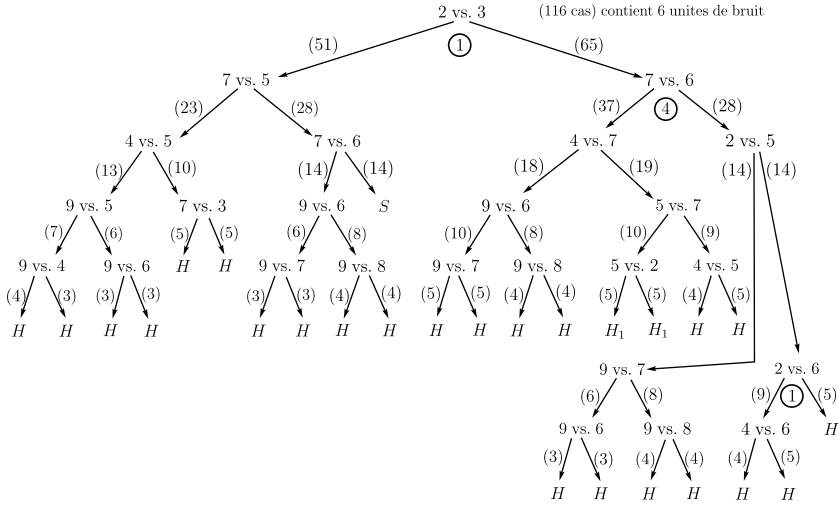
Suite de R_{E^*} pour $n = 9$ 

Figure 16

REMERCIEMENT

L'auteur souhaite remercier A. Hadian, de l'Université du Minnesota, pour les nombreuses conversations qui ont stimulé ces études. En particulier c'est lui qui reconnut que l'approche de (6.3) par la programmation dynamique pouvait être appliquée au problème du tri des n nombres.

BIBLIOGRAPHIE

- [1] R. C. BOSE, R. J. NELSON. A sorting problem. *J. ACM*, 9(2):282–296, Avril 1962.
- [2] W. H. BURGE. Sorting, trees, and measures of order. *Information and Control*, 1(3):181–197, 1958.
- [3] L. CARROLL. *Lawn Tennis Tournaments: The True Method of Assigning Prizes with a Proof of the Fallacy of the Present Method*. Macmillan, 1883.
- [4] Y. CESARI. *Questionnaire, codage et tris*. Problèmes de traitement de l'information. Centre Nat. de la Recherche Scient., Inst. Blaise Pascal, 1968.
- [5] H. A. DAVID. Tournaments and paired comparisons. *Biometrika*, 46(1-2):139–149, 06 1959.
- [6] H. A. DAVID. *The Method of Paired Comparisons*. Griffin's statistical monographs & courses. Hafner Publishing Company, 1963.
- [7] F. DUBAIL. *Algorithmes de questionnaires optimaux au sens de différents critères*. Université de Lyon, France, 1967.
- [8] L. R. FORD JR., S. M. JOHNSON. A tournament problem. *The American Mathematical Monthly*, 66(5):387–389, 1959.
- [9] W. A. GLENN. A comparison of the effectiveness of tournaments. *Biometrika*, 47(3-4):253–262, 12 1960.
- [10] A. HADIAN. Optimality properties of various procedures for ranking n different numbers using only binary comparisons. Technical Report 117, Department of Statistics, University of Minnesota, Minneapolis, MN, USA, 1969.
- [11] T. N. HIBBARD. Some combinatorial properties of certain trees with applications to searching and sorting. *J. ACM*, 9(1):13–28, Janvier 1962.
- [12] D. A. HUFFMAN. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.
- [13] K. E. IVERSON. A programming language. Dans *Proceedings of the May 1-3, 1962, Spring Joint Computer Conference*, AIEE-IRE '62 (Spring), page 345–351, New York, NY, USA, 1962. Association for Computing Machinery.
- [14] S. S. KISLITSYN. Sur l'estimation du plus petit nombre moyen de comparaisons par paires nécessaires à l'ordonnement complet de N objets de poids différents. *Vestnik Université de Leningrad. Série 1 : Mathématiques, mécanique et astronomie*, 18(1):162–163, 1962.

- [15] S. S. KISLITSYN. Affinement de l'estimation du plus petit nombre moyen de comparaisons nécessaires pour ordonner complètement la population finale. *Vestnik Université de Leningrad. Série 1 : Mathématiques, mécanique et astronomie*, 19(4):143–145, 1963.
- [16] R. J. MAURICE. Selection of the population with the largest mean when comparisons can be made only in pairs. *Biometrika*, 45(3-4):581–586, 12 1958.
- [17] C. F. PICARD. *Théorie des questionnaires*. Les Grands problèmes des sciences, 20. Gauthier-Villars, 1965.
- [18] M. SANDELIUS. On an optimal search procedure. *The American Mathematical Monthly*, 68(2): 133–134, 1961.
- [19] J. SCHREIER. O systemach eliminacji w turniejach. *Mathesis Polska*, 7:154–160, 1932.
- [20] J. ŚLUPECKI. On the systems of tournaments. *Colloquium Mathematicae*, 2(3-4):286–290, 1951.
- [21] M. SOBEL. Optimal group testing. Proc. Colloq. Inform. Theory, Debrecen 1967, 411–488, 1968.
- [22] M. SOBEL, P. A. GROLL. Group testing to eliminate efficiently all defective in a binomial sample. *The Bell System Technical Journal*, 38(5):1179–1252, 1959.
- [23] H. STEINHAUS. *Mathematical snapshots*. New York: Oxford University Press, 1950.
- [24] H. STEINHAUS. *One hundred problems in elementary mathematics*. Popular Lectures in Mathematics. Oxford etc.: Pergamon Press; Warszawa: PWN-Polish Scientific Publishers, 1963.
- [25] H. STEINHAUS. Some remarks about tournaments. Calcutta Mathematical Society, *Golden Jubilee Commemoration Volume (1958-1959)*, Partie 2, 323–327. Golden Jubilee Celebration Committee, 1959.